

For the past five years, *Personal Computer World* has led the market in microcomputer magazines and has developed a reputation for publishing the very best software for all the most popular micros. Now the cream of these programs has been gathered together in a single volume, enhanced and converted for the Spectrum.

Along with the forgotten gems from the archives available for the first time for Spectrum, this *Best of PCW* series also contains the finest programs written especially for this popular micro, plus a wealth of hints, tips and utilities which will prove essential reading for all serious programmers.

ISBN 0 7126 0264 X  
CENTURY PUBLISHING CO. LTD

BEST OF PCW

SOFTWARE FOR THE SPECTRUM



BEST OF PCW

# SOFTWARE FOR THE SPECTRUM



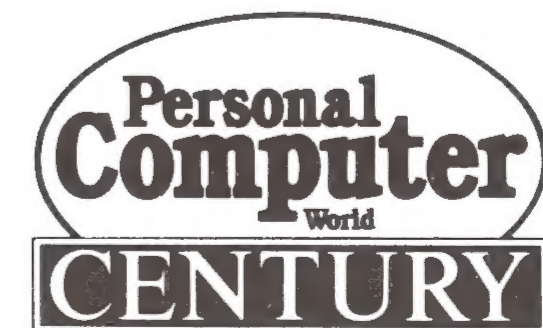


**BEST OF  
PERSONAL  
COMPUTER WORLD  
SOFTWARE FOR THE  
SPECTRUM**

## **Available in Century's Science & Technology series**

Dictionary of New Information Technology  
A. J. MEADOWS, M. GORDON and A. SINGLETON  
What to Buy for Business: A Handbook of New Office Technology  
JOHN DERRICK and PHILLIP OPPENHEIM  
The Electronic Mail Handbook  
STEPHEN CONNELL and IAN A. GALBRAITH  
The Way the New Technology Works  
KEN MARSH  
Mind  
DAVID A. TAYLOR  
Computers and Your Child  
RAY HAMMOND  
International Dictionary of Graphic Symbols  
JOEL ARNSTEIN  
*In association with 'Personal Computer World'*  
Microcomputing for Business: A User's Guide  
edited by DICK OLNEY  
The Microcomputer Handbook: A Buyer's Guide  
edited by DICK OLNEY  
The Spectrum Handbook  
TIM LANGDELL  
35 Programs for the Dragon 32  
TIM LANGDELL  
The Intimate Machine  
NEIL FRUDE  
35 Educational Programs for the BBC Micro  
IAN MURRAY  
Educational Programs for the Dragon 32  
IAN MURRAY  
Educational Programs for the Spectrum  
IAN MURRAY  
Century Computer Programming Course  
PETER MORSE and IAN ADAMSON  
Information Technology Yearbook  
edited by PHILIP HILLS  
Computer Gamesmanship  
DAVID LEVY  
Best of PCW Software for the BBC  
Best of PCW Software for the Dragon  
The Database Primer  
ROSE DEAKIN

# **BEST OF PERSONAL COMPUTER WORLD SOFTWARE FOR THE SPECTRUM**



**Century Communications  
London**

# CONTENTS

Introduction	9
Patience*	10
Sheepdog Trial	20
Quadrangle*	29
Virus	41
Planets	45
Target Practice	47
Showjumping	53
Cheese	61
Go-Moku*	67
Battleships	76
Duckshoot	82
Tarot*	89
Balloon	106
Fighter Pilot	114
Book Index	120
Zombies	124
Aircraft Lander*	130
Blaster	142
Stockmarket*	147
Rocket Attack	159
Truth*	166
Race Chase	186
Spec-Trek*	194
Convoy	213
Search and Rescue*	218
Hints and Tips	227

\* This program needs a 48K Spectrum

Copyright © Personal Computer World and  
Century Publishing Co. Ltd  
All rights reserved  
First published in Great Britain in 1983 by  
Century Communications Ltd (a subsidiary of Century  
Publishing Co. Ltd), Portland House, 12-13 Greek Street,  
London W1V 5LE

Reprinted 1984 (twice)

ISBN 0 7126 0264 X

Printed in Great Britain in 1984 by The Garden City Press Limited,  
Letchworth, Hertfordshire SG6 1JS.

## Acknowledgements

We would like to say a special thank you to our hard-working converters and referees. Also to Ms Jane Green for organising everyone so well. It has cost much midnight oil, maybe even a few premature grey hairs, and the high standard of this book owes much to their perseverance. Thanks also to Patrick Lynn for the inspired illustrations.

### Note

Throughout the book italic characters in the program listings represent graphics characters. To enter these press CAPS SHIFT and 9 together (to get graphics mode) and then the italic character. To get out of graphic mode press 9 again.

### Switching Off

Because previous programs can affect the way subsequently loaded programs run, it is always a wise precaution to switch off your micro before typing in or loading from tape any new programs. Some programs in this book will only work properly if this is done.



# INTRODUCTION

The Information Revolution is all about getting the information you want when you want it. How many times have you wished that you had all of PCW's programs and Spectrum hints in one volume?

For over five years PCW has been collecting the best in programs and programming tips for every micro. Now we've selected the finest of these and tailor-made the programs to run on your Spectrum computer.

Each program was redesigned by a Spectrum expert to make full use of colour, sound and the other facilities of your machine. All the programs have been closely vetted by one of PCW's experienced team of referees to ensure that they are as bug-free as possible.

For a pinch of spice we've added all the best readers' tips and programming hints, providing you with a unique compendium of the Best of PCW for your machine.

Of course one day we'll be inviting you to dial up this information but for the time being enjoy a good, old-fashioned book!

Jane Bird, editor  
Personal Computer World





ANDREW ESMOND  
Original program by Richard Still

## GENERAL DESCRIPTION

This rather complex program allows you to play one of the most well known and difficult games of patience on your Spectrum.

It demonstrates the use of user-defined graphics to represent the different suits of cards, and also shows how to manipulate a pack of cards on the computer, which can be used as a basis for writing many other card games yourself.

## DETAILED DESCRIPTION

Lines 10-48 Set up numeric and string variables.

50-70 Print 'shuffling pack' message.

80-100 Define the arrays.

110-190 Shuffle the pack of cards.

210-310 Print the initial version of the screen, showing the values of those cards which are face up.

320-410 Get input from keyboard. If the command is deal, move from pack or give up then act on it immediately.

420-460 If command was move from a column, set up the X and Y co-ordinates of the card(s) to be moved.

470-650 Input the column to which the card(s) are to be moved.

660-810 Moving from one column to another.

820-930 Moving to foundation.

940-960 Checking for errors and printing appropriate error messages.

970-1020 Determine suit and value of card in X\$.

1030-1080 Move the cursor to column X, line Y.

1100-1250 Check if you have won or lost; print congratulations or commiserations as appropriate. Then see if player wants to play again.

1260-1360 Deal a card.

1370-1399 Close up the remaining cards available for dealing when one has been taken into play.

1400-1610 Instructions routine.

1710-1810 Routine to finish the game automatically if no further progress is possible.

9000-9100 Set up user-defined graphics for the card suits and the foundation area.

```

10 REM ***Stilton Patience***
20 REM ***R.W.Still***
30 REM ***Initialize***
38 LET W1=0: LET L1=0
40 GO TO 1400
41 LET G=0: LET NU=1: LET X$=""
": LET S2=1: DIM V$(32)
45 LET Q$=""
46 LET X=1: LET T=0
47 LET H$=""
48 DIM Q(4)
50 RESTORE 160: CLS : PRINT TA
B 4: INVERSE 1: "
"
60 PRINT TAB 4; INVERSE 1; " I'
M SHUFFLING THE PACK "
70 PRINT TAB 4; INVERSE 1; "
"
80 LET L=0: DIM t(7): DIM b(7)

```



```

: DIM f$(13,2): DIM p$(52,2): DI
M s$(4): DIM n$(13): DIM a$(7,20
,2): FOR n=1 TO 7: LET t(n)=n: L
ET b(n)=n: NEXT n: LET l=1
  90 DIM f(4): DIM p(4): RANDOMI
ZE : LET p=0
  100 DIM e$(26)
  110 LET d$=""
  140 REM ***SHUFFLE PACK***
  150 FOR I=1 TO 4: READ S$(I): N
EXT I: FOR I=1 TO 13: READ N$(I)
: NEXT I
  160 DATA "B","D","A","C","A","2
","3","4","5","6","7","8","9","T
","J","Q","K"
  170 LET C$="ACABAAAD2C2B2A2D3C3
B3A3D4C4B4A4D5C5B5A5D6C6B6A6D7C7
B7A7D8C8B8A8D"
  180 LET C$=C$+"9C9B9A9DTCTBTATD
JCJBJAJDQCQBQAQDKCKBKAKD": LET C
=52
  190 FOR I=1 TO 52: LET A=INT (R
ND*C)+1: LET P$(I)=C$(2*A-1 TO 2
*A): LET L$=C$(1 TO 2*(A-1)): LE
T R$=C$(2*A+1 TO LEN C$): LET C$
=L$+R$: LET C=C-1: NEXT I
  210 REM ***PRESENT LAYOUT***
  220 CLS : FOR I=1 TO 7: FOR J=1
TO I: PRINT AT J,(I-1)*3;
  260 IF J=B(I) THEN PRINT P$(L)
: LET A$(I,J)=P$(L): LET L=L+1:
NEXT J: NEXT I: GO TO 280
  270 LET A$(I,J)=P$(L): PRINT I
NVERSE 1;" ": LET L=L+1: NEXT J
: NEXT I
  280 FOR I=29 TO 52: LET D$=D$+P
$(I): NEXT I
  290 PRINT AT 0,20:"E1111111111F
": FOR F=1 TO 13: PRINT AT F,20:
"J          K": NEXT F
  300 PRINT TAB 20:"G1111111111H"

```

```

310 INVERSE 1: FOR I=0 TO 18 ST
EP 3: PRINT AT 0,I;" ";STR$ (1+I
/3): NEXT I: INVERSE 0
  320 REM ***MAIN LOOP-FIRST INPU
T***
  330 IF D$<>"" THEN GO TO 360
  340 FOR I=1 TO 7: IF B(I)>1 THE
N LET I=8: NEXT I: GO TO 360
  350 NEXT I: GO SUB 1720
  360 LET PI=0: LET X=0: LET Y=0:
LET T=0: LET TT=0: PRINT AT 19,
0;TAB 24;AT 19,0;"FROM ";
  380 LET Z$=INKEY$: IF Z$="" THE
N GO TO 380
  390 IF Z$="D" THEN GO TO 1270
  400 IF Z$="E" THEN GO TO 1110
  410 IF Z$="P" THEN LET PI=1: G
O TO 480
  420 REM ***SET UP X AND Y***
  430 IF Z$>"7" OR Z$<"1" THEN G
O TO 460
  440 LET T=VAL Z$: LET X=T: LET
Y=B(X)
  450 IF Y=0 THEN GO TO 330
  460 IF T=0 THEN GO TO 380
  470 REM ***SECOND INPUT***
  480 PRINT Z$;" TO ";
  490 LET B$=INKEY$: IF B$="" THE
N GO TO 490
  491 IF B$="F" THEN GO TO 500
  492 IF B$<"1" OR B$>"9" THEN G
O TO 490
  500 IF B$="F" AND X=0 THEN GO
TO 360
  505 IF B$="F" THEN LET TT=8: G
O TO 520
  510 LET TT=VAL B$
  520 IF TT=0 THEN GO TO 490
  530 IF TT=T THEN GO TO 490
  540 PRINT B$;
  550 IF TT=8 THEN LET Y=T(X)

```



```

560 IF PI=1 THEN LET X$=H$: GO
TO 580
570 LET X$=A$(X,Y)
580 GO SUB 970
590 IF TT=8 THEN GO TO 830
600 IF T(TT)=0 AND NU<>13 THEN
GO TO 330
610 IF T(TT)=0 THEN LET B(TT)=
1: GO TO 670
620 LET S1=SU: LET N1=NU
630 LET X$=A$(TT,T(TT))
640 GO SUB 970
650 IF NU-1<>N1 OR 1-SU<>S1 THE
N GO TO 950
660 REM ***MAIN MOVE ROUTINE***
670 GO SUB 1040: LET F=1
680 IF PI=1 THEN LET F=2: LET
F$(1)=H$: GO SUB 1380: LET P=P-1
: GO TO 740
690 FOR I=B(X) TO T(X): LET F$(
F)=A$(X,I): LET A$(X,I)=" ": L
ET F=F+1
700 PRINT " ": OVER 1;V$( TO 3
0);: NEXT I: LET XX=X: LET YY=Y
710 LET B(X)=B(X)-1: LET T(X)=B
(X)
720 IF B(X)>0 THEN LET G=1
730 IF T(TT)=0 THEN LET X=TT:
LET Y=0: GO SUB 1040: GO TO 750
740 LET X=TT: LET Y=T(TT): GO S
UB 1040
750 PRINT OVER 1;V$;: FOR I=1
TO F-1: PRINT F$(I): OVER 1;V$(
TO 30);: LET A$(X,T(X)+1)=F$(I)
760 IF F$(I)<>" " THEN LET T(X
)=T(X)+1
770 NEXT I
780 FOR I=1 TO 12: LET F$(I)="
": NEXT I: LET F=1
790 IF G=1 THEN LET G=0: LET X
=XX: LET Y=YY-1: GO SUB 1040: PR

```

```

INT A$(X,Y)
800 IF PI=1 THEN GO TO 1310
810 GO TO 330
820 REM ***MOVE TO FOUNDATION**
*
830 IF F(S2)<>NU-1 THEN GO TO
330
840 LET Q(S2)=Q(S2)+1: LET F(S2
)=NU
850 PRINT AT F(S2),20+(S2*2);
860 IF Y=1 THEN GO TO 880
870 IF A$(X,Y-1)<>" " THEN LE
T G=1
880 PRINT X$
890 IF PI=1 THEN GO SUB 1380:
LET P=P-1: GO TO 1310
900 GO SUB 1040: PRINT " ": L
ET A$(X,Y)=" ": IF Y>0 THEN LE
T T(X)=T(X)-1
910 IF T(X)=B(X)-1 THEN LET B(
X)=B(X)-1
920 IF G=1 THEN LET G=0: PRINT
AT (24-PEEK 23689)-1,33-PEEK 23
688-2;A$(X,Y-1)
930 GO TO 330
940 REM ***ERROR ROUTINE***
950 IF 1-SU<>S1 THEN PRINT AT
19,0;TAB 24;AT 19,0;"WRONG COLOU
R";: FOR I=1 TO 200: NEXT I: GO
TO 330
960 PRINT AT 19,0;TAB 24;AT 19,
0;"WRONG VALUES";: FOR I=1 TO 20
0: NEXT I: GO TO 330
970 REM ***TO RETURN VALUE AND
NUMBER OF X$***
980 RESTORE 160: FOR I=1 TO 4:
READ W$: IF X$(LEN X$)=W$ THEN
LET SU=I-INT (I/2)*2: LET S2=I
990 NEXT I
1000 FOR I=1 TO 13: READ W$: IF
X$(1)=W$ THEN LET NU=1: LET I=1

```



```

4
1010 NEXT I
1020 RETURN
1030 REM ***MOVE CURSOR TO X,Y**
*
1040 LET X1=X: LET Y1=Y: PRINT A
T 0,0;
1050 LET X1=X1*3-3: IF X1=0 THEN
GO TO 1070
1060 GO TO 1080
1070 IF Y1=0 THEN RETURN
1080 PRINT AT Y1,X1;: RETURN
1100 REM ***'END' ROUTINE***
1110 FOR I=1 TO 4: IF Q(I)<>13 T
HEN GO TO 1160
1120 NEXT I: PRINT AT 6,0; FLASH
1;"YOU'VE WON!!"
1121 LET W1=W1+1
1130 FOR I=1 TO 1000: NEXT I
1140 GO TO 1190
1160 PRINT AT 6,0; FLASH 1;"SORR
Y, YOU'VE LOST"
1161 LET L1=L1+1
1170 FOR I=1 TO 300: NEXT I
1190 CLS : PRINT "____": P
RINT INVERSE 1;" SCORES ____": PR
INT : PRINT "WINS:";W1: PRINT :
PRINT "LOSSES:";L1
1210 PRINT : PRINT "PLAY AGAIN?(
Y/N)"
1220 LET Z$=INKEY$: IF Z$="" THE
N GO TO 1220
1230 IF Z$="Y" THEN GO TO 41
1240 IF Z$<>"N" THEN GO TO 1220
1250 PRINT "GOODBYE!!!": STOP
1260 REM ***DEAL ROUTINE***
1270 LET P=P+3
1280 IF D$<>" " THEN GO TO 1310
1290 FOR I=1 TO 7: IF B(I)>1 THE
N LET I=8: NEXT I: GO TO 1330
1300 NEXT I: GO SUB 1720

```

```

1310 IF P=0 THEN LET H$=" ": G
O TO 1360
1320 IF LEN D$/2=P-1 THEN LET P
=LEN D$/2: GO TO 1350
1330 IF LEN D$/2=P-2 THEN LET P
=LEN D$/2: GO TO 1350
1340 IF P>LEN D$/2 THEN LET P=0
: GO TO 1270
1350 LET H$=D$(2*P-1 TO (2*P-1)+
1)
1360 PRINT AT 21,0;H$: GO TO 330
1370 REM ***CLOSE UP REMAINING C
ARDS***
1380 LET L$=D$( TO (P-1)*2): LET
R$=D$(2*P+1 TO LEN D$): LET D$=
L$+R$: RETURN
1399 STOP
1400 BORDER 0: PAPER 0: INK 6: C
LS
1401 GO SUB 9000
1410 PRINT "_____"
"
1420 PRINT INVERSE 1;" *
**STILTON*** "
1430 PRINT " FROM 1-7 CARDS AR
E DEALT TO SEVEN COLUMNS,THE LA
ST CARD ONLY"
1440 PRINT "OF EACH COLUMN BEING
FACE UPWARD-S"
1450 PRINT " EXPOSED CARDS ARE
MOVED IN"
1460 PRINT "ASCENDING SUIT SEQUE
NCE TO ACES AS THESE APPEAR,AND
IN DESCENDI-NG";
1470 PRINT " SEQUENCE OF ALTERNA
TE COLOURS TO THE BOTTOM CARDS OF
COLUMNS."
1480 PRINT " COMPLETE SEQUENCES
OF CARDS MAYBE MOVED BETWEEN COL
UMNS."
1490 PRINT "EMPTY COLUMNS MAY BE

```



```

FILLED ONLY BY SEQUENCES HEADED
BY KINGS."
1500 PRINT "    THE REST OF THE P
ACK IS DEALT, THREE CARDS AT A TI
ME TO A    WASTE"
1510 PRINT "PILE, THE TOP CARD OF
WHICH IS    ALWAYS AVAILABLE. ANY
NUMBER OF    REDEALS ARE ALLOWED.
"
1520 PRINT "PRESS 'ENTER' TO CON
TINUE."
1530 INPUT  LINE I$: CLS
1540 PRINT "    CARDS ARE MOVED BY
SIMPLY    TYPING THE ORIGINAL
COLUMN NUMB-ER, OR P FOR WASTE PI
LE, FOLLOWED BY THE DESTINATION C
OLUMN"
1550 PRINT "NUMBER OR 'F' FOR FO
UNDATION."
1560 PRINT : PRINT " D DEALS A F
RESH GROUP OF THREE CARDS, E ENDS
THE GAME"
1570 PRINT : PRINT "PLEASE PRES
S 'ENTER'."
1580 INPUT  LINE A$: POKE 23658,
8
1590 DIM F(4)
1600 FOR F=1 TO 4: LET F(F)=13:
NEXT F
1610 GO TO 41
1710 REM ***AUTOMATIC FINISH***
1720 FOR Q=1 TO 7: IF T(Q)<>0 TH
EN LET Q=8: NEXT Q: GO TO 1740
1730 NEXT Q: GO TO 1110
1740 FOR Q=1 TO 7: IF T(Q)=0 THE
N NEXT Q: GO TO 1720
1742 LET X=Q: LET Y=T(Q)
1745 LET X$=A$(X,Y): GO SUB 970
1750 IF F(S2)<>NU-1 THEN \NEXT Q
: GO TO 1720
1760 LET Q(S2)=Q(S2)+1: LET F(S2

```

```

)=NU: PRINT AT 0,20;
1770 PRINT AT F(S2),20+(S2*2);
1780 PRINT X$
1790 GO SUB 1040: PRINT "    ";: L
ET A$(X,Y)="    ": IF Y>0 THEN LE
T T(X)=T(X)-1
1800 IF T(X)=B(X) THEN LET B(X)
=B(X)-1
1810 NEXT Q: RETURN
8999 STOP
9000 RESTORE 9020: FOR I=0 TO 7
9010 READ A,B,C,D: POKE USR "A"+
I,A: POKE USR "B"+I,B: POKE USR
"C"+I,C: POKE USR "D"+I,D: NEXT
I
9020 DATA 0,BIN 1000100,16,16,16
,BIN 11101110,BIN 111000,BIN 111
000
9030 DATA BIN 00111000,254,BIN 1
111100,BIN 1010100
9040 DATA BIN 01111100,254,BIN 1
1111110,BIN 11111110,BIN 1111111
0,BIN 1111100,BIN 10010010,BIN 0
1010100,BIN 01111100,BIN 111000,
16,16
9050 DATA BIN 00111000,16,16,16,
16,0,BIN 111000,16
9060 FOR F=0 TO 7: READ A,B,C,D:
POKE USR "E"+F,A: POKE USR "F"+
F,B: POKE USR "G"+F,C: POKE USR
"H"+F,D: NEXT F
9070 DATA 0,0,16,8,0,0,16,8,0,0,
16,8,0,0,8,16,7,224,7,224,8,16,0
,0,16,8,0,0,16,8,0,0
9080 FOR F=0 TO 7: POKE USR "I"+
F,0: NEXT F: POKE USR "I"+4,255
9090 FOR F=0 TO 7: POKE USR "J"+
F,16: POKE USR "K"+F,B: NEXT F
9100 RETURN

```



# SHEEPDOG TRIAL

LEON GOODFRIEND  
Original program by Simon Williams

## GENERAL DESCRIPTION

It's just one man and his dog who have the task of penning the three sheep in this sheepdog trial simulation. You must guide your obedient and loyal dog around the four progressively more difficult stages to force the three obstinate sheep into their pens. Various natural and manmade obstacles make your task harder and you lose points the longer it takes. On completion of the four stages your score is given for the sheepdog trial.

## DETAILED DESCRIPTION

Lines 10-65 Print the title page and set up the user-defined graphics characters.  
70-240 Display the instructions.  
250-360 Draw course for stage one.  
390-410 Position sheep and dog.  
420-495 Stage one.  
500 Auto-scroll.  
510-560 End page for stage one.  
570-730 Draw stage two course.  
740-750 Locate dog and sheep.  
760-805 Stage two.  
810 Auto-scroll.  
820-880 Print stage two end page.  
900-1060 Display the stage three course.  
1070-1090 Position three sheep and the dog.  
1100-1160 Stage three.  
1170-1240 The end page for stage three.  
1250-1350 Draw the fourth stage course.  
1360-1410 Position sheep and dog and draw the pen.

1420-1460 Stage four.

1465-1600 Play congratulatory tune, flash the border, print the final page, ask user for another go.

7000-7030 Introductory tune.

8000-8240 User control of the sheepdog and semi-random movement of the three sheep.

9000-9070 The initial starting positions of the three sheep and the sheepdog.

```
10 PAPER 7: INK 0: FLASH 0: BR
IGHT 0: OVER 0: INVERSE 0: BORDE
R 0: CLS
20 PRINT INVERSE 1;AT 2,13;"S
HEEP";AT 4,14;"DOG";AT 6,13;"TRI
AL"
30 FOR a=0 TO 31
40 READ b: POKE USR "a"+a,b
50 NEXT a
60 DATA 0,0,127,255,91,17,17,0
,0,0,65,254,63,33,33,0,8,28,28,6
2,62,127,8,28,0,129,255,129,255,
129,255,129
65 GO SUB 7000
70 PRINT ""In this game you m
ust make"
80 PRINT "your sheepdog B guid
e three"
90 PRINT "sheep A through a se
ries of"
100 PRINT "obstacles."
110 PRINT ""Controls are:-"
120 PRINT TAB 12;"T Y U"
130 PRINT TAB 12;"G H"
140 PRINT TAB 12;"V B N"
150 FOR a=0 TO 2*PI STEP PI/4
160 PLOT 123,28: DRAW 12*COS a,
9*SIN a
170 NEXT a
180 PRINT ""There are four st
ages to the"
190 PRINT "test. Points are los
```



```

t for taking"
200 PRINT "too long over a stag
e. The sheep"
210 PRINT "will only move when
the dog gets"
215 PRINT "close to them"
216 PRINT "In the first three s
tages you must chase the sheep
into the space at the base of
the screen."
217 PRINT "Each sheep becomes i
nactive whenin its pen"
218 PRINT "In stage two the blu
e areas are ponds which the shee
p will not cross,but if there a
re gaps in the fencing between
the ponds, the sheep will go th
rough them."
220 PRINT "In stage four all th
e sheep willremain active until
the third one is penned."
225 DIM x(3): DIM y(3): LET p=0
230 PRINT "'Press any key to s
tart"
240 PAUSE 0
250 PAPER 4: CLS
260 PRINT "'TAB 8;"DDDDDDDDDDDDDDDD
DDDD"
270 PRINT TAB 8;"D      oooooo
D"
280 FOR a=4 TO 12: PRINT AT a,8
;"D";TAB 23;"D": NEXT a
290 PLOT 73,110
300 FOR a=1 TO 12
310 DRAW 9,0,-PI+RND-.5
320 NEXT a
330 PRINT AT 7,12;"CC  CC";AT
8,12;"CC  CC"
340 PRINT AT 5,19;"/\"
360 PRINT AT 11,10;"####
^"
370 PRINT AT 13,8;"DDDDDDDD  DD

```

```

DDDD"
380 PLOT 117,62: DRAW 27,0,1.5*
PI
390 GO SUB 9000
400 PRINT AT 18,0;"Stage score:
";ss;" "
410 GO SUB 8000
420 FOR a=1 TO n
430 IF y(a)<=13 THEN GO TO 480
440 LET y(a)=y(n): LET x(a)=x(n
): LET n=n-1
450 BEEP .3,0
480 NEXT a
490 IF n>0 THEN GO TO 400
495 FOR a=1 TO 200: NEXT a
500 POKE 23692,255
510 FOR a=1 TO 25: PRINT : NEXT
a
520 PRINT TAB 6;"Stage one comp
leted"
530 PRINT "'TAB 11;"Score: ";ss
,.....
540 LET p=ss
545 GO SUB 7000
550 PRINT "'Press any key for
stage two".....
560 IF INKEY$="" THEN GO TO 56
0
570 CLS
590 PLOT 71,60: DRAW 0,100,-PI:
DRAW 115,0: DRAW 0,-100,-PI
600 PLOT 70,60: DRAW 0,100,-PI:
DRAW 115,0: DRAW 0,-100,-PI
610 DRAW -33,0: DRAW -20,-40: D
RAW -15,0: DRAW -20,40: DRAW -32
,0
620 DRAW 0,70
625 INK 1
630 FOR x=80 TO 200 STEP 24
640 FOR a=0 TO 7
650 LET b=INT (SQR (49-a*a)+.5)
660 PLOT x+a,100-b: DRAW 0,2*b

```



```

670 PLOT x-a,100-b: DRAW 0,2*b
680 NEXT a
690 NEXT x
700 INK 0
710 FOR a=11 TO 23 STEP 3
720 IF RND<.7 THEN PRINT AT 9,
a;"D"
730 NEXT a
740 GO SUB 9000
745 PRINT AT 20,0;"Running scor
e: ";p;"Stage score: ";
748 PRINT AT 21,13;ss;" ";
750 GO SUB 8000
760 FOR a=1 TO n
770 IF y(a)<15 THEN GO TO 790
780 LET x(a)=x(n): LET y(a)=y(n
): LET n=n-1
785 BEEP .3,0
790 NEXT a
800 IF n>0 THEN GO TO 748
803 GO SUB 7000
805 FOR a=1 TO 200: NEXT a
810 POKE 23692,255
820 FOR a=1 TO 25: PRINT : NEXT
a
830 PRINT TAB 6;"Stage two comp
leted"
840 PRINT ""Score so far: ";ss
+p
850 LET p=p+ss
860 PRINT ""Press any key for
stage three"
870 PRINT "....."
880 IF INKEY$="" THEN GO TO 88
0
900 CLS
910 PRINT ""TAB 6;"CCCCCCCCCCCC
CCCCCCCC"
920 PRINT TAB 6;"CCCCCCCCCC
CCC"
930 PRINT TAB 6;"CC
CCC"

```

```

940 PRINT TAB 6;"C          CC
C      C"
950 PRINT TAB 6;"CC          CCCC
C      CC"
960 PRINT TAB 6;"CCCC          CCCCC
CC"
970 PRINT TAB 6;"CCCCCCCCCCCCC
D"
980 PRINT TAB 6;"C          CCCCCC
D"
990 PRINT TAB 6;"C          CCC
DDD"
1000 PRINT TAB 6;"CCC      ^^
DD"
1010 PRINT TAB 6;"CCCC      ^^   CCC
CDD"
1020 PRINT TAB 6;"CCCCC      CCCCCC
CD"
1030 PRINT TAB 6;"CCCC          CCCCC
C"
1040 PRINT TAB 6;"CCCC          CCCCC
C"
1050 PRINT ""Running score: ";p
1060 PRINT "Stage score:"
1070 GO SUB 9000
1080 PRINT AT 19,13;ss;" "
1090 GO SUB 8000
1100 FOR a=1 TO n
1110 IF y(a)<15 THEN GO TO 1130
1120 LET y(a)=y(n): LET x(a)=x(n
): LET n=n-1
1125 BEEP .3,0
1130 NEXT a
1140 IF n>0 THEN GO TO 1080
1150 GO SUB 7000
1160 FOR a=1 TO 200: NEXT a
1170 CLS
1180 PRINT ""TAB 5;"Stage three
completed"
1190 PRINT ""Score: ";ss+p
1200 LET p=p+ss
1210 PRINT ""Now you just have

```



```

to get the"
1220 PRINT "sheep into their pen
"
1230 PRINT """"Press any key"
1240 PAUSE 0
1250 CLS
1260 PRINT "" CDDDCDDDCDDDCDDDC
CDDDCDDDCDDDC"
1270 FOR a=3 TO 16
1280 IF a-3*INT (a/3)=2 THEN PR
INT AT a,2;"C";TAB 30;"C": GO TO
1300
1290 PRINT AT a,2;"D";TAB 30;"D"
1300 NEXT a
1310 PRINT " CDDDCDDDCDDDCDDDCD
DDCDDDCDDDC"
1320 FOR a=1 TO 20
1330 LET x=RND*25+4: LET y=RND*1
2+4
1340 PRINT AT y,x;CHR$ (146+RND)
1350 NEXT a
1360 GO SUB 9000
1370 PRINT AT 19,0;"Running scor
e: ";p;"Stage score:"
1390 PRINT AT 10,13;" ";AT 11
,13;" ";AT 12,13;" "
1400 PLOT 104,73: DRAW 0,23: DRA
W 31,0: DRAW 0,-23
1405 PRINT AT 20,13;ss;" "
1410 GO SUB 8000
1420 LET ct=0
1430 FOR a=1 TO 3
1440 IF x(a)>12 THEN IF x(a)<17
THEN IF y(a)>9 THEN IF y(a)<1
3 THEN LET ct=ct+1
1450 NEXT a
1460 IF ct<>3 THEN GO TO 1405
1465 GO SUB 7000
1470 FOR a=1 TO 5
1480 FOR b=7 TO 0 STEP -1
1485 BORDER b
1490 PAUSE 10

```

```

1500 NEXT b
1510 NEXT a
1520 POKE 23692,255
1530 FOR a=1 TO 25: PRINT : NEXT
a
1540 PRINT "You have now complet
ed the trial"
1550 PRINT ""Your dog scored ";
ss+pi" points out of"
1560 PRINT "a possible 80."
1570 PRINT """"Do you wish to
play again -y/n "; FLASH 1;"?"
,,,,,,
1580 IF INKEY$="y" OR INKEY$="Y"
THEN RUN
1590 IF INKEY$="n" OR INKEY$="N"
THEN GO TO 9999
1600 GO TO 1580
7000 FOR a=10 TO 25: BEEP .06,a:
NEXT a
7010 FOR a=10 TO 25: BEEP .06,a:
NEXT a
7015 FOR a=1 TO 20: NEXT a
7020 FOR a=25 TO 10 STEP -1: BEE
P .06,a: NEXT a
7030 RETURN
8000 LET nm=nm+1
8010 LET a=20-INT ((nm-50)/40)
8020 IF a>20 THEN GO TO 8040
8030 LET ss=a: IF ss<0 THEN LET
ss=0
8040 LET a$=INKEY$
8050 LET px=x+(a$="u" OR a$="h"
OR a$="n")-(a$="t" OR a$="g" OR
a$="v")
8060 LET py=y-(a$="t" OR a$="y"
OR a$="u")+ (a$="v" OR a$="b" OR
a$="n")
8070 IF SCREEN$ (py,px)=" " THEN
GO TO 8120
8080 IF SCREEN$ (py,x)=" " THEN
LET px=x: GO TO 8120

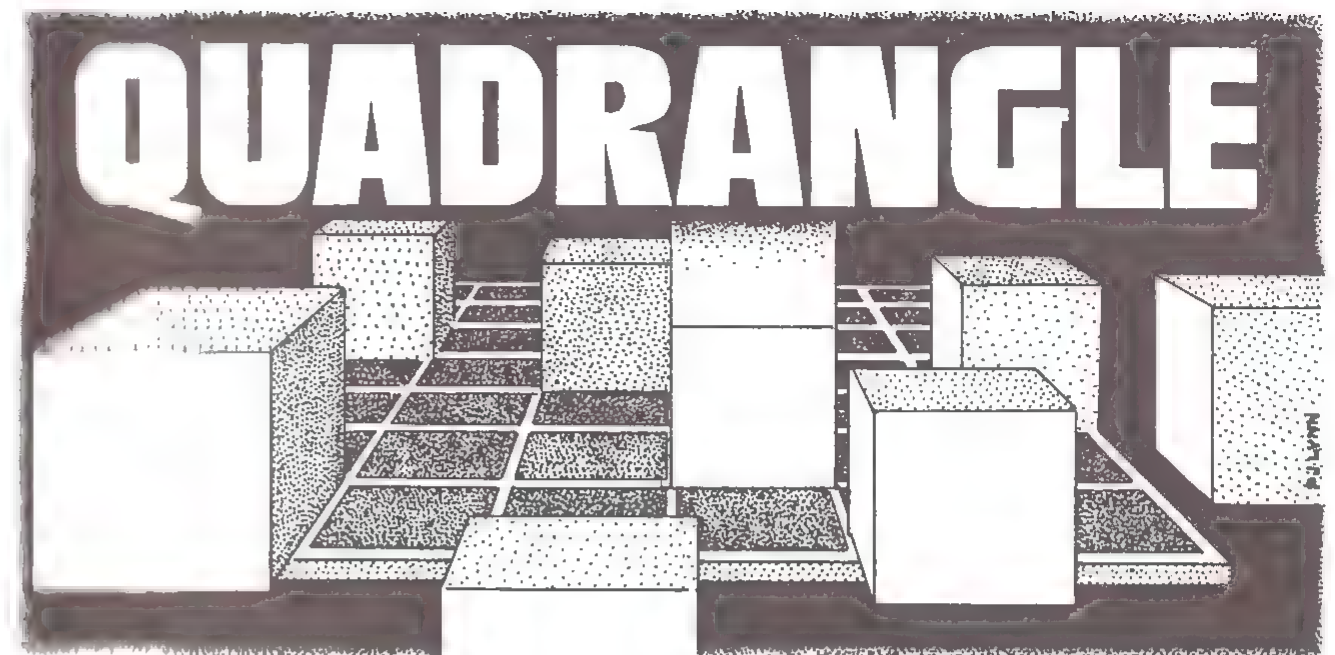
```



```

8070 IF SCREEN$ (y,px)=" " THEN
  LET py=y: GO TO 8120
8100 LET px=x: LET py=y
8120 PRINT AT y,x;" ";AT py,px;"
  B"
8130 LET x=px: LET y=py
8140 FOR a=1 TO n
8150 LET dx=x(a)-x: LET dy=y(a)-
  y: LET d=dx*dx+dy*dy
8160 IF d>9 THEN GO TO 8230
8165 IF d=9 AND n<3 THEN GO TO
  8230
8170 LET px=x(a)+SGN dx: LET py=
  y(a)+SGN dy
8180 IF SCREEN$ (py,px)=" " THEN
  GO TO 8220
8190>IF SCREEN$ (py,x(a))=" " THE
  N LET px=x(a):GO TO 8220
8200 IF SCREEN$ (y(a),px)=" " TH
  EN LET py=y(a): GO TO 8220
8210 IF RND>.01 THEN LET px=x(a
  )+SGN (RND-.5): LET py=y(a)+SGN
  (RND-.5): GO TO 8180
8215 LET px=x(a): LET py=y(a)
8220 PRINT AT y(a),x(a);" "; INK
  7;AT py,px;"A"
8225 LET x(a)=px: LET y(a)=py
8230 NEXT a
8240 RETURN
9000 LET ss=20: READ x,y: LET nm
  =0: LET n=3
9010 FOR a=1 TO 3: READ x(a),y(a
  ): PRINT AT y(a),x(a); INK 7;"A"
  : NEXT a
9020 PRINT AT y,x;"B"
9030 RETURN
9040 DATA 10,3,12,6,14,4,16,5
9050 DATA 6,12,4,8,6,8,6,9
9060 DATA 8,5,10,4,11,6,14,4
9070 DATA 5,5,18,11,5,14,24,6

```



LEON GOODFRIEND

Original program by

Eileen Baghoomians and Steven Fawthrop

## GENERAL DESCRIPTION

In this challenging tactical game your task is to position your counters on the board so that they form the four corners of a square. The game can be played by two players or against the computer, which has three levels of difficulty. The program shows how a computer can make logical decisions about where to place its pieces, based on a formula for calculating the relative value of different positions, similar to the methods used in writing chess programs.

## DETAILED DESCRIPTION

Lines 10-60 Set up machine code and user-defined graphics.  
 110-160 Define arrays and set up their contents.  
 170-212 Give instructions and find out how many players.  
 213-300 Ask for level of play and find out who starts.  
 310 Set up board.  
 320 Print out whose move.  
 330-334 Is it the computer's first move? If so, decide its starting column randomly.



340-370 See what player's move is, and check that it's legal.  
 380 Print recommended move if required.  
 390-395 Move player's or computer's piece to appropriate column.  
 400-410 Print piece in position on the screen.  
 420-480 Has someone made a quadrangle?  
 520-650 Alter board according to move.  
 660-764 Game is over: is it a draw or has someone won?  
 See if player wants to play again.  
 780-1000 Decide the computer's move.  
 1050-1080 Print out the board.  
 1090-1160 Set of short utility subroutines.  
 1400 Data.  
 8000 Clear bottom of screen.  
 9000-9800 Print the instructions.

```

10 INK 0: PAPER 7: FLASH 0: BR
IGHT 0: OVER 0: INVERSE 0: BORDE
R 0: CLS
15 PRINT AT 11,10; FLASH 1;"Pl
ease wait"
20 CLEAR 60000
25 PRINT AT 11,10; FLASH 1;"Pl
ease wait"
30 FOR a=60004 TO 60018: READ
b: POKE a,b: NEXT a
35 REM data for machine code A
ND routine
40 DATA 33,96,234,126,35,35,16
6,71,43,126,35,35,166,79,201
50 FOR a=0 TO 7: READ b: POKE
USR "a"+a,b: NEXT a
60 DATA 126,129,189,189,189,18
9,129,126
110 DIM a$(2,12): DIM l(2): DIM
d$(2,3): DIM s(2): DIM f(71,3):
DIM g(13): DIM m(8,8,3)
111 LET b$=""
120 LET d$(1)=CHR$ 16+CHR$ 0+"A
"
```

```

130 LET d$(2)=d$(1): LET d$(2,2
)=CHR$ 7
140 FOR k=2 TO 71: READ f(k,1):
NEXT k: LET g(1)=1: FOR k=2 TO
13: LET g(k)=2*g(k-1): NEXT k
170 GO SUB 9000
175 PRINT INVERSE 1;AT 7,11;"Q
UADRANGLE"
180 INPUT "One or two players?"
"1 LINE p$
184 IF CODE p$>57 OR CODE p$<49
THEN GO TO 180
185 LET p=VAL p$: IF p<>1 AND p
<>2 THEN GO TO 180
200 IF p=1 THEN INPUT "Your na
me please ";x$: LET y$="Computer
"
210 IF p=2 THEN INPUT "First p
layer's name? ";x$;"Second playe
r's name? ";y$: LET p=0
211 LET l(1)=LEN x$: LET l(2)=L
EN y$: LET a$(1)=x$: LET a$(2)=y
$: IF l(1)>12 THEN LET l(1)=12
212 IF l(2)>12 THEN LET l(2)=1
2
213 CLS : PRINT AT 5,10; INVERS
E 1;"YOUR PIECES""
214 FOR k=1 TO 2: PRINT 'a$(k,
TO l(k)),d$(k): NEXT k
220 IF p=1 THEN INPUT "Level o
f play (1-3) ? "; LINE t$: IF CO
DE t$>57 OR CODE t$<49 THEN GO
TO 220
221 IF p<>1 THEN GO TO 230
225 LET t=VAL t$: IF t<>1 AND t
<>2 AND t<>3 THEN GO TO 220
230 IF p=0 THEN LET t=0: FOR a
=1 TO 500: NEXT a
240 LET ts=t: LET s(1)=0: LET s
(2)=0
250 LET rp1=INT (RND*2.5+3.5):
LET rn1=INT (RND*2.5+2.5): LET r
```

```

m=50: LET zx=0: CLS
260 IF a$(2)<>"Computer" THEN
  GO TO 310
270 IF t<3 THEN LET p=1: GO TO
  310
280 INPUT "Do you want to go fi
rst -y/n ";b$: IF b$<>"y" AND b$
<>"n" AND b$<>"N" AND b$<>"Y" TH
EN GO TO 280
290 LET p=b$="n"
300 IF b$="y" OR b$="Y" THEN L
ET zx=1
310 FOR k=1 TO 7: FOR l=2 TO 7:
  LET m(k,l,1)=-1: NEXT l: LET m(
k,l,1)=0: FOR m=2 TO 3: FOR l=1
TO 7: LET m(k,l,m)=0: NEXT l: NE
XT m: NEXT k: GO SUB 1050
320 GO SUB 8000: PRINT AT 21,0;
a$(p+1, TO 1(p+1));"'s move";: L
ET rm=rm+1
330 IF p<>1 OR a$(2)<>"Computer
" THEN GO TO 340
331 IF zx=1 THEN LET q=p: GO T
O 780
332 LET zx=1: LET g=INT (RND*4)
+3: IF g>=6 THEN LET g=4+(INT (
RND*2)+2)*SGN (RND-.5)
334 GO TO 350
340 BEEP .8,5: INPUT "Column (1
-7) ? "; LINE b$: IF (CODE b$>57
OR CODE b$<49) AND b$<>"h" THEN 340
345 IF b$<>"h" THEN LET g=VAL
b$
350 IF b$="h" THEN LET q=0: LE
T t=3: GO TO 780
360 IF g<1 OR g>7 THEN PRINT A
T 18,0;"There is no such column
- pleasetry again": FOR k=0 TO 5
00: NEXT k: GO TO 320
370 LET g=g-1: IF m(g+1,1,1)=6
THEN PRINT AT 18,0;"Column ";g+
1;" is full - please try again":

```

```

FOR k=0 TO 500: NEXT k: GO TO 3
20
380 LET t=ts: IF b$="h" THEN P
RINT AT 18,0;TAB 31;" I recommen
d ";g+1;". ": GO TO 340
390 LET m(g+1,1,1)=m(g+1,1,1)+1
: LET m(g+1,m(g+1,1,1)+1,1)=p: G
O SUB 400: GO SUB 420: IF e$<>"y
" THEN GO SUB 520
395 GO TO 660
400 IF p=1 AND a$(2)="Computer
" THEN PRINT AT 15-2*m(g+1,1
,1),2*g+10; FLASH 1;d$(p+1): FOR
l=5 TO 15: BEEP .2,1: NEXT l
410 PRINT AT 15-2*m(g+1,1,1),2*
g+10;d$(p+1)
420 LET q=p: LET x=g: LET y=m(g
+1,1,1)
430 LET e$="n"
440 FOR a=1 TO 6: IF a=y THEN
GO TO 470
445 IF x+y-a>6 OR x+y-a<0 THEN
GO TO 455
450 IF m(x+1,a+1,1)=q AND m(x+y
-a+1,y+1,1)=q AND m(x+y-a+1,a+1,
1)=q THEN LET e$="y": LET y8=1:
GO TO 480
455 IF x-y+a>6 OR x-y+a<0 THEN
GO TO 470
460 IF m(x+1,a+1,1)=q AND m(x-y
+a+1,y+1,1)=q AND m(x-y+a+1,a+1,
1)=q THEN LET e$="y": LET y8=-1
: GO TO 480
470 NEXT a
480 RETURN
520 LET v1=2^x+2^(y+6): POKE 60
000,INT (v1/256): POKE 60001,v1-
256*PEEK 60000
522 FOR b=2 TO 71: POKE 60002,I
NT (f(b,1)/256): POKE 60003,f(b,
1)-256*PEEK 60002: LET vm=USR 60
004: IF vm=v1 THEN GO SUB 540

```



```

530 NEXT b: RETURN
540 IF p=1 THEN GO TO 600
542 IF ABS (f(b,2))>5 THEN RETURN
543 IF f(b,2)<0 THEN GO TO 620
544 LET f(b,2)=f(b,2)+1: LET f0=7: IF f(b,2)=1 OR f(b,2)=2 THEN LET f0=f(b,2)
550 LET u5=1
560 POKE 60000,INT (f(b,1)/256): POKE 60001,f(b,1)-256*PEEK 60000: FOR u=0 TO 12: POKE 60002,INT (g(u+1)/256): POKE 60003,g(u+1)-256*PEEK 60002: LET vx=USR 60004: IF vx=0 THEN GO TO 580
570 IF u<7 AND u<>x THEN LET u1=u
572 IF u>6 AND u<>y THEN LET u2=u-6
580 NEXT u: LET m(x+1,u2+1,u5+1)=m(x+1,u2+1,u5+1)+f0: LET m(u1+1,y+1,u5+1)=m(u1+1,y+1,u5+1)+f0: LET m(u1+1,u2+1,u5+1)=m(u1+1,u2+1,u5+1)+f0
590 RETURN
600 IF ABS f(b,2)>5 THEN RETURN
602 IF f(b,2)>0 THEN GO TO 640
603 LET f(b,2)=f(b,2)-1: LET f0=7: IF f(b,2)=-1 OR f(b,2)=-2 THEN LET f0=-f(b,2)
610 LET u5=2: GO TO 560
620 LET f0=-7: IF f(b,2)=-1 OR f(b,2)=-2 THEN LET f0=f(b,2)
630 GO SUB 610: LET f(b,2)=10: RETURN
640 LET f0=-7: IF f(b,2)=1 OR f(b,2)=2 THEN LET f0=-f(b,2)
650 GO SUB 550: LET f(b,2)=-10: RETURN
660 FOR g=1 TO 7: IF m(g,1,1)<>6 THEN GO TO 680

```

```

665 NEXT g
670 IF e$<>"y" THEN GO SUB 800
680 PRINT AT 18,6: FLASH 1:"The game is a draw": GO TO 750
690 IF e$<>"y" THEN LET p=1-p: GO TO 320
700 LET s(p+1)=s(p+1)+1: GO SUB 8000: PRINT AT 19,13-1(p+1)/2;a$(p+1, TO 1(p+1)):" wins"
710 PRINT OVER 1: FLASH 1: INK 0: AT 15-2*y,10+2*x;" ";AT 15-2*a,10+2*x;" ";AT 15-2*y,10+2*(x+y*(y-a));" ";AT 15-2*a,10+2*(x+y*(y-a));" "
720 FOR a=1 TO 4 STEP 3: FOR b=1 TO 3: BEEP .3+.2*(b=3),a+4: NEXT b: NEXT a: FOR a=1 TO 5: BEEP .3+.5*(a=5),11-ABS (3-a): NEXT a
730 FOR a=1 TO 500: NEXT a
740 PRINT AT 20,0:: FOR a=1 TO 20: PRINT " ";a$(a, TO 1(a)),s(a): NEXT a
750 INPUT "Another game -y/n ?": IF e$<>"y" AND e$<>"n" AND e$<>"N" AND e$<>"Y" THEN GO TO 750
760 IF e$="n" OR e$="N" THEN GO TO 9999
762 INPUT "Are the players the same ";e$: IF e$="n" OR e$="N" THEN RUN
763 IF e$="y" OR e$="Y" THEN GO TO 250
764 GO TO 762
780 PRINT AT 18,0: FLASH 1:"Thinking";
785 FOR k=1 TO 7: PRINT FLASH 1;".":: BEEP .02,0
790 IF m(k,1,1)=6 THEN LET m(k,1,2)=-1000: GO TO 810
800 LET r=k-1: LET s=m(k,1,1)+1

```

```

: GO SUB 1090: LET m(k,1,2)=vx
810 NEXT k: GO SUB 990: IF m(g+
1,1,2)>998 THEN GO TO 380
820 FOR k=1 TO 7: IF m(k,1,1)<5
THEN LET q=1-p: LET r=k-1: LET
s=m(k,1,1)+2: GO SUB 1090: IF v
x=1000 THEN LET m(k,1,2)=-999
830 IF m(k,1,2)>-999 THEN PRIN
T FLASH 1;".": BEEP .02,0: LET
q=p: LET r=k-1: LET s=m(k,1,1)+
1: GO SUB 1100: LET m(k,1,2)=vx
840 NEXT k
850 IF t=1 THEN GO TO 960
852 IF t=2 THEN GO SUB 1120: G
O TO 960
860 FOR k=1 TO 7: IF m(k,1,2)<-
997 THEN GO TO 950
862 LET m(k,m(k,1,1)+2,1)=p: LE
T m(k,1,1)=m(k,1,1)+1
870 FOR m=1 TO 7: IF m(m,1,1)=6
THEN LET m(m,1,3)=-1000: GO TO
880
875 LET q=1-p: LET r=m-1: LET s
=m(m,1,1)+1: GO SUB 1100: LET m(
m,1,3)=vx
880 NEXT m: PRINT FLASH 1;".":
: BEEP .02,0
890 FOR a=1 TO 4: LET l=-100: F
OR m=1 TO 7: IF m(m,1,3)>1 THEN
LET l=m(m,1,3): LET lq=m
900 NEXT a: LET m(lq,1,3)=-1000
: LET m(k,1,2)=m(k,1,2)-l*rn1
910 NEXT m
920 LET q=p: LET r=k-1: LET s=m
(k,1,1)+1: GO SUB 1090: IF vx=10
00 THEN LET m(k,1,2)=m(k,1,2)-r
n1*rm
930 LET m(k,1,1)=m(k,1,1)-1: LE
T m(k,m(k,1,1)+2,1)=-1
940 PRINT FLASH 1;".": BEEP .
02,0
950 NEXT k: LET q=p

```

```

970 GO SUB 990
980 GO TO 380
990 LET g=-1: LET l=-1000: FOR
k=1 TO 7: IF m(k,1,2)>1 THEN LE
T l=m(k,1,2): LET g=k-1: GO TO 1
000
995 IF m(k,1,2)=1 THEN IF ABS
(4-k)<ABS (4-g) THEN LET l=m(k,
1,2): LET g=k-1
1000 NEXT k: RETURN
1050 FOR a=0 TO 6: PLOT 76+16*a,
155: DRAW 0,-96: PLOT 91+16*a,15
0: DRAW 0,-96: NEXT a
1055 FOR a=0 TO 5: PLOT 76,59+16
*a: DRAW 111,0: PLOT 76,76+16*a:
DRAW 111,0: NEXT a
1060 PRINT AT 0,0;a$(1, TO 1(1))
;" ";s(1),a$(2, TO 1(2));" ";s(2
)
1070 FOR a=1 TO 7: PRINT AT 15,8
+2*a;a: NEXT a
1080 RETURN
1090 LET vx=0: LET x=r: LET y=s:
GO SUB 430: IF e$="y" THEN LET
vx=1000: RETURN
1092 LET q=1-q: LET x=r: LET y=s
: GO SUB 430: LET q=1-q: IF e$="
y" THEN LET vx=999
1094 RETURN
1100 LET vx=m(k,m(k,1,1)+2,1+p):
RETURN
1120 FOR k=1 TO 7: IF m(k,1,2)<-
997 THEN GO TO 1160
1122 LET m(k,m(k,1,1)+2,1)=1: LE
T m(k,1,1)=m(k,1,1)+1
1130 IF m(k,1,1)<6 THEN LET q=1
-p: LET r=k: LET s=m(k,1,1)+1: G
O SUB 1100: LET m(k,1,2)=m(k,1,2
)-rp1*vx
1140 LET m(k,1,1)=m(k,1,1)-1: LE
T m(k,m(k,1,1)+2,1)=-1
1150 LET q=p+2: LET r=k: LET s=m

```



```

(k,1,1)+1: GO SUB 1100: LET m(k,
1,2)=m(k,1,2)+rp1*vx
1160 PRINT FLASH 1;".": NEXT k
: LET q=p: RETURN
1400 DATA 387,390,396,408,432,48
0,771,774,780,792,816,864,1539,1
542,1548,1560,1584,1632,3075,307
8,3084,3096,3120,3168,6147,6150,
6156,6168,6192,6240,645,650,660,
680,720,1285,1290,1300,1320,1360
,2565,2570,2580,2600,2640,5125,5
130,5140,5160,5200,1161,1170,118
8,1224,2313,2322,2340,2376,4617,
4626,4644,4680,2193,2210,2244,43
69,4386,4420,4257,4290
8000 FOR c=18 TO 21: PRINT AT c,
0;TAB 31;" ";; NEXT c: RETURN
9000 REM instructions
9005 INK 7
9010 FOR a=0 TO 21: PRINT AT a,0
;"
";: NEXT a
9020 FOR a=0 TO 64 STEP 8
9030 INK a/8
9040 PLOT a,a: DRAW 255-2*a,0: D
RAW 0,175-2*a: DRAW 2*a-255,0: D
RAW 0,2*a-175
9050 BEEP .1,a/4
9060 NEXT a
9070 INK 0
9080 FOR a=0 TO 50
9090 POKE 23606,50-a
9100 PRINT AT 11,11;"QUADRANGLE"
9110 BEEP .06,a
9120 NEXT a
9130 INPUT "Do you want instruct
ions -y/n ?";i$
9140 IF i$<>"y" AND i$<>"n" AND
i$<>"N" AND i$<>"Y" THEN GO TO
9130
9150 IF i$="n" OR i$="N" THEN G
O TO 9800

```

```

9160 POKE 23692,255
9170 PRINT AT 21,0
9180 PRINT "'TAB 11;"QUADRANGLE
"
9190 PRINT "'At the beginning o
f the game you"
9200 PRINT "will be given the op
tion to play"
9210 PRINT "against the computer
or against"
9220 PRINT "another player."
9230 PRINT "The object of the g
ame is to put"
9240 PRINT "four pieces at the
corners of a"
9250 PRINT "square. The first t
o do this is"
9260 PRINT "the winner."
9270 PRINT "Each column is fill
ed from the"
9280 PRINT "bottom up."
9290 PRINT "'If you choose to
play against"
9300 PRINT "the computer you ha
ve a choice"
9310 PRINT "of three levels of d
ifficulty of"
9320 PRINT "which level three is
hardest."
9330 PRINT '#0;"Press any key to
continue"; FLASH 1;"_";
9340 PAUSE 0
9345 IF INKEY$<>" " THEN GO TO 9
345
9350 PRINT #0
9360 PRINT "When it is your turn
you can get"
9370 PRINT "help from the comput
er: type 'h'"
9380 PRINT "and the computer wil
l recommend"
9390 PRINT "a move. This option

```

```

is available"
9400 PRINT "even when not play
ing against"
9410 PRINT "the computer. When
two people"
9420 PRINT "are playing the game
can be made"
9430 PRINT "faster by removing '
GO SUB 520'"
9440 PRINT "from line 390. Howe
ver you will"
9450 PRINT "not receive any mean
ingful help"
9460 PRINT "if you type 'h'."
9470 PRINT ""
9480 PRINT #0;"Press any key to
play"; FLASH 1;"_";
9490 PAUSE 0
9500 PAPER 4: CLS : RETURN

```

# VIRUS

JOHN WALKER

Original program by Allan and Sue Vining

## GENERAL DESCRIPTION

The object of this game is to show the user how to write real time games on the Spectrum.

In this game the object is to trap the virus, shown as a 'V', using your antibody, shown as an 'A', by forcing the virus into a position where it cannot move without either running into a wall, its own trail or your trail.

## DETAILED DESCRIPTION

Lines 1220-1240 Get keyboard input, validate it and move the antibody in the appropriate direction. Check whether or not the antibody crashes into a wall or a virus.

1250-1268 Is the virus completely trapped?

1270-1295 Is the virus's present direction of travel blocked? If so, change the direction of travel to one which is not blocked.

1300-1310 Move the virus.

1320 If at least one virus is still not trapped then repeat the main program loop.

1330-1360 All destroyed so print congratulatory message, increase the number of virus by one and if it is less than six, go through the game again. If it is six, the game is over so stop.

1370 You have crashed, so print message and stop.

2000-3500 Display instructions.

4000-4200 Set up virus x, y co-ordinates and direction, and antibody x,y co-ordinates.

4300-4600 Print out the board and initial positions of the virus and antibody.

5000-6000 Sets up user-defined characters.



```

1000 RANDOMIZE : DIM a(10,3): LE
T n=1
1120 GO SUB 5000
1140 GO SUB 2000
1160 GO SUB 4100
1220 LET a$=INKEY$: IF CHR$ COD
E a$<"5" OR CHR$ CODE a$>"8" THE
N LET A$=L$
1225 LET L$=A$
1230 LET dx=(a$="8")-(a$="5"): L
ET dy=(a$="6")-(a$="7")
1235 IF SCREEN$ (ay+dy,ax+dx)<>"
" THEN GO TO 1370
1240 LET ax=ax+dx: LET ay=ay+dy:
PRINT INVERSE 1; INK 7;AT ay,a
x;"A"
1250 FOR i=1 TO n: IF a(i,3)=0 T
HEN GO TO 1310
1260 IF SCREEN$ (a(i,1)+1,a(i,2)
)=" " THEN GO TO 1270
1262 IF SCREEN$ (a(i,1)-1,a(i,2)
)=" " THEN GO TO 1270
1264 IF SCREEN$ (a(i,1),a(i,2)+1
)=" " THEN GO TO 1270
1266 IF SCREEN$ (a(i,1),a(i,2)-1
)=" " THEN GO TO 1270
1268 LET a(i,3)=0: GO TO 1310
1270 IF RND>.9 THEN GO TO 1290+
5*(RND>.5)
1280 LET dx=(a(i,3)=1)-(a(i,3)=3
): LET dy=(a(i,3)=2)-(a(i,3)=4):
IF SCREEN$ (a(i,1)+dy,a(i,2)+dx
)=" " THEN GO TO 1300
1285 IF RND>.5 THEN GO TO 1295
1290 LET a(i,3)=a(i,3)+1-4*(a(i,
3)=4): GO TO 1280
1295 LET a(i,3)=a(i,3)-1+4*(a(i,
3)=1): GO TO 1280
1300 LET a(i,1)=a(i,1)+dy: LET a
(i,2)=a(i,2)+dx: PRINT AT a(i,1)
,a(i,2); INK 6;"A"
1310 NEXT i

```

```

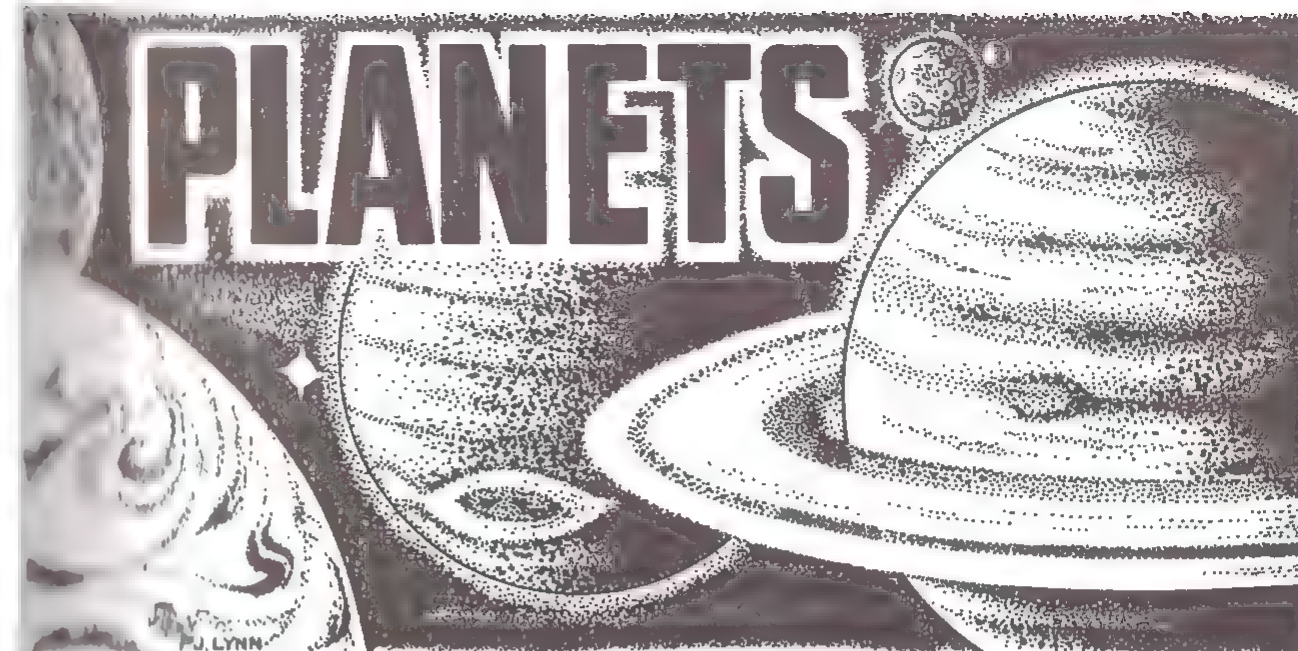
1320 FOR i=1 TO n: IF a(i,3)<>0
THEN GO TO 1220
1323 NEXT i
1330 CLS : PRINT AT 11,0;"You ha
ve destroyed this screen's";AT 1
1,7;"worth of virus."
1335 FOR P=1 TO 15: BEEP .1,P: N
EXT P
1340 LET n=n+1: IF n<6 THEN PAU
SE 200: GO TO 1160
1350 PRINT ''TAB 9;"You have won
!"
1360 FOR P=15 TO 1 STEP -1: BEEP
.1,P: NEXT P: RUN
1370 CLS : PRINT AT 11,7;"You ha
ve crashed!": GO TO 1360
2000 REM Virus
2010 BORDER 0: PAPER 0: INK 7: C
LS : PRINT AT 2,13; FLASH 1;"VIR
US"AT 2,13; OVER 1;"_____"
2020 PRINT '"You control an anti
body "'A"' that"
2030 PRINT "must trap a virus "'
V"', by making"
2040 PRINT "the virus move into
a position"
2050 PRINT "where it has no empt
y spaces to"
2060 PRINT "move into. As you e
liminate"
2070 PRINT "each screen's worth
of virus, a"
2080 PRINT "new batch forms for
you to"
2090 PRINT "destroy."
3000 PRINT '"Use the arrows key
s (5-8) to"
3100 PRINT "control the movement
of your"
3200 PRINT "antibody."
3300 PRINT '' FLASH 1;"Press any
key to start."

```

```

3310 BEEP 1,1
3400 PAUSE 0
3500 RETURN
4100 CLS : FOR i=1 TO n: LET a(i
,1)=1+INT (RND*20): LET a(i,2)=1
+INT (RND*30): LET a(i,3)=1+INT
(RND*4): NEXT i
4200 LET ay=11: LET ax=15
4300 INVERSE 1: PRINT "SCORE:";n
;"XXXXXXXXXXXXXXXXXXXXXXXXX"
4400 FOR i=1 TO 20: PRINT AT i,0
;"X";AT i,31;"X": NEXT i
4500 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX": INVERSE 0
4600 LET L$=CHR$ (53+RND*3): PRI
NT INVERSE 1; INK 7;AT ay,ax;"A
": FOR i=1 TO n: PRINT AT a(i,1)
,a(i,2); INK 6;"A": RETURN
5000 FOR i=0 TO 7: READ a: POKE
USR "a"+i,a: NEXT i: RETURN
6000 DATA 165,66,60,90,90,60,66,
165

```



LEON GOODFRIEND  
Original program by C. Ward

## GENERAL DESCRIPTION

This program recreates the orbits of the four inner planets of our solar system. You can observe Mercury (M), Venus, the Earth, and Mars happily circling the Sun. Their speeds and orbital radii have been scaled down so that they do not wander off the edges of your TV screen.

NOTE:- For ease of programming the planets follow circular rather than their true elliptical orbits.

## DETAILED DESCRIPTION

Lines 10-50 Set initial colours and display instructions.  
60-70 Set up the user-defined character.  
80-150 Store the data for each planet in arrays.  
160-200 Work out the new position for each planet in turn.  
210-270 Display the four planets and the Sun, then loop back to determine their new positions.



```

10 INK 6: PAPER 0: FLASH 0: BRIGHT 1:
INVERSE 0: BORDER 0: CLS
20 PRINT "''''''This program shows the
relative"
30 PRINT "positions of the four inner
most"
40 PRINT "planets of our Solar System
and"
50 PRINT "the Sun.": PAUSE 200
60 FOR a=0 TO 7: READ b: POKE USR "a"+
a,b: NEXT a
70 DATA 60,126,255,255,255,255,126,60
80 DIM t(4): DIM d(4): DIM r(4): DIM p
(4,2): DIM c(4): DIM n$(4,5): DIM l(4)
90 FOR a=1 TO 4
100 READ t(a),d(a),r(a),c(a),a$: LET n$
(a)=a$: LET l(a)=LEN a$
110 NEXT a
120 DATA 327.95626,0.5240327,20,2,"Mars
"
130 DATA 0,0.98561,13.180432,1,"Earth"
140 DATA -258.47927,1.6021291,9.5330396
,7,"Venus"
150 DATA -135.14322,4.0923507,5.1013216
,6,"M"
160 FOR a=1 TO 4
170 LET p(a,1)=30+SIN (t(a)/180*PI)*r(a
)
180 LET p(a,2)=20+COS (t(a)/180*PI)*r(a
)
190 LET t(a)=t(a)-d(a)
200 NEXT a
210 CLS
220 PRINT AT 10,15;"*Sun"
230 FOR a=1 TO 4
240 PRINT INK c(a); OVER 1;AT p(a,2)/2
,p(a,1)/2;"a"; INK 6;n$(a, TO l(a))
250 NEXT a
260 BEEP .6,20
270 GO TO 160

```

# TARGET PRACTICE

J. BINNINGTON  
Original program by Gordon Mills

## GENERAL DESCRIPTION

This game consists of one or two players trying to destroy the targets on the screen by firing a missile from a constantly moving missile launcher at the bottom of the screen. Each player has fifteen shots and his score is displayed at the bottom of the screen as the game progresses. The game will provide a test of eye-hand co-ordination and, with its nine levels of difficulty, will not be easily mastered.

## DETAILED DESCRIPTION

100-20 Set up array which contains positions of targets and the number of shots each player has.  
30-50 Print initial screen picture.  
60-110 Print the instructions.  
120-255 Input both players' names and the speed of the missile base for each player.  
260-370 Fill the array A with the positions on the screen of ten targets in random positions for each player.  
380-500 Print the border between the two players' sectors and the line at the bottom of the screen.  
510-535 Set score to 0 and call the subroutine to set up the missile's shape in the user-defined graphics.  
540 Main program loop is repeated a maximum of 150 times.  
545 If both players have fired all their missiles then go to the 'game over' routine.  
550 If player 1 has fired all his missiles then skip to player 2.  
560 Set y co-ordinate of missile launcher.

570-580 Print missile launcher at each position in player 1's sector.  
 590 Waiting loop whose length depends on the speed chosen by player 1.  
 600 Delete the missile launcher from its present position on the screen.  
 610-616 See if player 1 has fired.  
 620 Call fire missile subroutine.  
 630 Reduce number of missiles left, check to see if all targets have been destroyed, if so give bonus score.  
 640-660 Increase score, print score, continue moving missile launcher.  
 700-820 Repeat lines 550-660 for player 2.  
 830-890 Game is over, so ask if the players want to play again, and if so RUN the program again.  
 1000-1070 Print the missile as it moves up the screen.  
 1080 If the missile has missed, then return from the subroutine.  
 1085 Calculate the score and BEEP.  
 1090 Delete the target from the array A and RETURN.  
 2000-2050 Set up the shape of the missile in the user-defined graphics.

```

10 REM Conversion of a Gordon
    Mills program by
    J.S.Binnington(1983).
20 DIM A(31): LET SN=15: LET S
M=15
30 PAPER 2: INK 6: BORDER 0
40 PRINT AT 11,8;"TARGET PRACT
ICE"
50 PAUSE 200
60 CLS
70 PRINT : PRINT " This is a g
ame for two players."
80 PRINT : PRINT " The object
of the game is to destroy as
many TARGETS as you can wi
th your 15 shots."
90 PRINT : INK 4: PRINT " The

```

```

player on the LEFT uses the lett
er 'A' the player on the RIGH
t uses the letter 'L' as"
100 PRINT " fire buttons. "
110 INK 6: PRINT : PRINT " The
speed of the missile laun
cher may be varied between fast
(1) to slow(9). "
120 REM *****
130 REM Get some details.
140 REM *****
150 PAUSE 150
160 INPUT "Enter the name of th
first player ";N$
165 IF LEN N$>10 THEN LET N$=N
$(1 TO 10)
170 PRINT : PRINT : PRINT "
";CHR$ 18+CHR$ 1;"Hi ";N$;; FLA
SH 0
180 INPUT "Enter the name of th
second player ";M$
185 IF LEN M$>10 THEN LET M$=M
$(1 TO 10)
190 PRINT : PRINT : PRINT "
";CHR$ 18+CHR$ 1;"Hi ";M$;; FL
ASH 0
200 INPUT "Enter speed(1-9) ple
ase "+N$+" ";A$
205 IF CODE A$<49 OR CODE A$>58
OR LEN A$>1 THEN : BEEP .2,0: G
O TO 200
210 LET A=CODE A$-48
220 PRINT AT 17,15;"Speed ";A
230 INPUT "Enter speed(1-9) ple
ase "+M$+" ";B$
235 IF CODE B$<49 OR CODE B$>58
OR LEN B$>1 THEN BEEP .5,0: GO
TO 230
240 LET B=VAL B$
250 PRINT AT 19,15;"Speed ";B

```



```

255 PAUSE 200
260 REM *****
270 REM Set up 10 random target
sfor each player
280 REM *****
290 CLS
295 FOR I=15 TO 17: LET A(I)=-1
: NEXT I
300 FOR L=1 TO 10
310 LET M=1+INT (RND*16)
320 IF A(M)<>0 THEN GO TO 310
330 LET I=INT (RND*10)+1
340 LET A(M)=I: LET A(M+17)=I
350 PRINT AT I,M;"#"
360 PRINT AT I,M+17;"*"
370 NEXT L
380 REM *****
390 REM Print the border+the sc
ore line
400 REM *****
410 LET Y=25: LET NS=0: LET MS=
0: LET NH=0: LET MH=0
420 FOR X=0 TO 255
430 PLOT X,Y: PLOT X,Y+1
440 NEXT X
450 LET X=127
460 FOR Y=0 TO 175
470 PLOT X,Y
480 NEXT Y
490 PRINT AT 19,1;N$: PRINT AT
21,1;"SCORE ";AT 21,11;NS
500 PRINT AT 19,18;M$: PRINT AT
21,18;"SCORE ";AT 21,27;MS
510 REM *****
520 REM Game proper starts here
530 REM *****
535 LET SC=0: GO SUB 2000
540 FOR I=1 TO 150
545 IF SN<1 AND SM<1 THEN LET
I=150: GO TO 820
550 IF SN<1 THEN GO TO 700
560 LET Y =17

```

```

570 FOR X=1 TO 14
580 PRINT AT Y,X;CHR$ 144
590 FOR D=1 TO A: NEXT D
600 PRINT AT Y,X;" "
610 LET G$=INKEY$: IF G$="" THE
N GO TO 660
614 IF G$="A" THEN GO TO 620
616 IF G$<>"a" THEN GO TO 660
620 LET R=A: GO SUB 1000
630 LET SN=SN-1: IF SC<>0 THEN
LET NH=NH+1: IF NH=10 THEN PRI
NT AT 19,8;"BONUS": LET SC=SC+(S
N*100): LET SN=0
640 LET NS=NS+SC: LET SC=0
650 PRINT AT 21,11;NS
660 NEXT X
670 IF SM<1 THEN GO TO 545
680 FOR X=18 TO 31
690 PRINT AT Y,X;CHR$ 144
700 FOR D=1 TO B: NEXT D
710 PRINT AT Y,X;" "
720 LET G$=INKEY$: IF G$="" THE
N GO TO 810
725 IF G$="L" THEN GO TO 770
730 IF G$<>"l" THEN GO TO 810
740 LET R=B: GO SUB 1000
750 LET SM=SM-1: IF SC<>0 THEN
LET MH=MH+1: IF MH=10 THEN PRI
NT AT 19,24;"BONUS": LET SC=SC+(
100*SM): LET SM=0
760 LET MS=MS+SC: LET SC=0
770 PRINT AT 21,27;MS
780 NEXT X
790 NEXT I
800 REM *****
810 REM Game over routine
820 REM *****
830 INPUT "Do you want to play
again(Y/N)? ";G$
840 IF G$="Y" THEN RUN
850 IF G$="y" THEN RUN
860 CLS : PRINT AT 11,10;"GOODB

```

YE ALL"

```
890 PAUSE 100: GO TO 880
1000 REM *****
1010 REM Fire subroutine
1020 REM *****
1030 LET SC=0: LET YY=Y
1040 FOR Y=17 TO A(X) STEP -1
1050 PRINT AT Y,X;CHR$ 144
1055 FOR D=1 TO 5: NEXT D
1060 PRINT AT Y,X;" "
1070 NEXT Y
1080 IF Y=-1 THEN GO TO 1090
1085 LET SC=10*INT (10-Y/R): BEE
P .1,-20
1090 LET A(X)=0: LET Y=YY: RETUR
N
2000 FOR U=0 TO 7
2010 READ Z
2020 POKE USR (CHR$ 144)+U,Z
2030 NEXT U
2040 DATA 24,24,24,24,60,90,219,
219
2050 RETURN
```

# SHOWJUMPING

LEON GOODFRIEND

Original program by Roy Bowden

## GENERAL DESCRIPTION

The simulation 1-4 horses and riders compete against the clock to jump-off. Enter the names of the riders and horses, then each in turn must try to complete the course in as quick a time as possible, and gaining the least amount of faults. You must guide your horse around the course, and when you reach a fence a close up view of it is depicted on the screen. You must now quickly judge its height and width and enter the appropriate number. If you have judged it correctly your horse will clear the fence, if not, you get four faults. At any time your horse may refuse to jump a fence, in which case you get three faults, and must try to jump the fence again. When all competitors have completed the course the winner is announced.

## DETAILED DESCRIPTION

Lines 10-29 Print title page and instructions.  
30-80 Set up the user-defined graphics characters.  
90-230 Initialisation and user input for number of competitors, their names and horses.  
240-390 Main program loop for each competitor and print the course.  
400-465 Start of the competitor's turn.  
490-530 Flash appropriate fence.  
540-690 Draw a close-up view of the fence.  
700-750 User input for fence height and width.  
760-850 Fence is successfully jumped.  
1000-2140 Horse did not clear the fence.  
2200-2250 Horse refused to jump the fence.  
2500-2530 Check for end of round.



3000-3040 The horse retired.  
 3100-3120 Horse is disqualified.  
 3500-3530 Result is displayed.  
 3550-4010 Print the details of the best round so far.  
 4020-8000 User input for another game.  
 9000-9090 Move horse under user control.  
 9300-9380 Print the fences.  
 9997-9999 Update the clock.

```

10 PAPER 7: INK 1: FLASH 0: BR
IGHT 0: OVER 0: INVERSE 0: BORDE
R 7: CLS
20 PRINT AT 1,9: FLASH 1: PAPE
R 6: "SHOW JUMPING"
21 PRINT AT 3,19: " K":AT 4,0: "
Movement keys are Z+X":AT 5,19:
" M"
22 PRINT AT 7,0: "First guide t
he horse (signified by the jockey
's cap) onto the start - the f
lashing "; FLASH 1: "S"
23 PRINT "Then ride over each
fence in turn. The next fence
to jump will be flashing."
24 PRINT "When you reach a fen
ce the display shows a clos
e-up view of it, with three possi
ble widths and three heights."
25 PRINT : PRINT
26 PRINT : PRINT "Enter a numb
er from 1 to 9 as follows..."
27 PRINT "
": PRINT " High Mediu
m Low ": PRINT "
"
28 PRINT " Wide 9 8
7 ": PRINT "
": PRINT " Medium 6
5 4 "
29 PRINT "
": PRINT " Narrow 3 2

```

```

1 ": PRINT "
"
30 FOR a=0 TO 39
40 READ b: POKE USR "a"+a,b
50 NEXT a
60 DATA 0,0,12,18,33,33,255,0
70 DATA 192,192,255,192,248,19
9,199,248,3,3,255,3,31,227,227,3
1
80 DATA 255,255,148,148,148,10
0,100,100,100,100,100,148,148,14
8,255,255
90 LET v=1
100 RANDOMIZE
110 INPUT "How many competitors
(1-4) ":p
120 IF p<1 OR p>4 THEN GO TO 1
10
130 DIM r$(p,10): DIM h$(p,20):
DIM p(15,3): DIM j(15,2): DIM f
(p): DIM t(p)
140 FOR c=1 TO p
150 INPUT ("Competitor ";c;" Na
me "):r$(c)"Horse's name ":h$(c
)
160 NEXT c
170 PRINT """"Please wait whi
le the jumps are checked."
180 FOR j=1 TO 15
190 READ p(j,1),p(j,2),p(j,3)
200 LET j(j,1)=FN r(3)
210 LET j(j,2)=FN r(3)
220 NEXT j
230 DATA 2,16,1,2,12,1,2,9,1,2,
0,1,5,1,2,10,1,2,15,1,2,21,1,2,2
0,6,1,16,10,1,12,15,1,16,19,2,22
,19,2,26,16,1,26,10,1
240 FOR c=1 TO p
250 CLS
260 PRINT ""Competitor number
":c
270 PRINT "r$(c)" "Riding ":h$(c)

```

```

280 PAUSE 150
300 GO SUB 9300
380 PRINT AT 20,2; PAPER 2; INK
7;"S"
390 BEEP 1,9
400 LET x=10: LET y=8: LET j=0
410 GO SUB 9000
420 IF a$="S" THEN BEEP 1,0: G
O TO 460
430 PRINT AT 19,0; INK 0; PAPER
7;"You can't jump any fences un
til you have been through the st
art."
440 BEEP 1,10: BEEP 1,5
450 GO TO 300
460 PRINT AT 20,2;" "
470 LET st=INT FN t()
480 LET j=1
485 PRINT AT 0,16;"Time: 0"
490 OVER 1: FLASH 1
500 PRINT AT p(j,2),p(j,1);" ";
510 IF p(j,3)=1 THEN PRINT " "
: GO TO 530
520 PRINT AT p(j,2)+1,p(j,1);"
"
530 FLASH 0: OVER 0
540 PRINT AT 0,22;INT FN t()-st
550 GO SUB 9000
560 IF ATTR (y1,x1)<128 THEN G
O TO 3100
600 BEEP .4,12: CLS
610 PLOT 3+15*j(j,1),10
620 DRAW 30,0: DRAW 0,30+20*j(j
,2): DRAW -30,0: DRAW 0,-30-20*j
(j,2)
630 PLOT 223-15*j(j,1),10
640 DRAW 30,0: DRAW 0,30+20*j(j
,2): DRAW -30,0: DRAW 0,-30-20*j
(j,2)
650 FOR a=20*j(j,2)-8 TO 28+20*
j(j,2) STEP 12
660 FOR b=0 TO 4

```

```

670 PLOT 33+15*j(j,1),a+b: DRAW
100-30*j(j,1),0
680 NEXT b
690 NEXT a
700 PRINT #0;"Enter number now"
705 BEEP .1,12
710 PAUSE 100
720 LET a$=INKEY$
730 IF a$="" THEN GO TO 2000
735 BEEP .1,12
740 IF INT ((FN r(15)+FN r(15))
/2)=j THEN GO TO 2200
750 IF CODE a$<49 OR CODE a$>57
THEN GO TO 2000
760 LET a=VAL a$
770 LET w=INT ((a-1)/3)
780 LET h=a-3*w
790 IF 3-w<>j(j,1) OR h<>j(j,2)
THEN GO TO 2000
800 PAPER 2: CLS
810 PRINT AT 4,8; FLASH 1;"Fenc
e cleared."
820 BEEP .4,5: BEEP .4,10: BEEP
.4,15
830 LET j1=j
840 LET j=j+1
850 GO TO 2500
1000 DEF FN r(x)=INT (RND*x)+1
2000 PAPER 2: INK 0: CLS
2010 PRINT ""You did not clear
the fence."
2020 PRINT "Penalty: 4 faults"
2030 LET f(c)=f(c)+4
2035 LET j1=j
2040 LET j=j+1
2050 BEEP .25,0: BEEP 1,-5
2100 PRINT ""You now have ";f(
c);" faults."
2110 PRINT #0;"Do you wish to re
tire y/n ?";
2115 LET a$=INKEY$
2120 IF a$="y" OR a$="Y" THEN G

```



```

0 TO 3000
2130 IF a$<>"n" AND a$<>"N" THEN
  GO TO 2115
2140 GO TO 2500
2200 PAPER 2: INK 0: CLS
2210 PRINT ""Your horse just r
efused."
2220 PRINT "Penalty: 3 faults"
2225 LET j1=j
2230 LET f(c)=f(c)+3
2240 BEEP 1,20
2250 GO TO 2100
2500 IF j>15 THEN GO TO 3500
2510 GO SUB 9300
2520 LET x=p(j1,1)+2: LET y=p(j1
,2)
2530 GO TO 490
3000 PAPER 6: INK 1: CLS
3010 PRINT ""h$(c)""ridden by
";r$(c)""TAB 12;"RETIRED"
3020 LET f(c)=100
3030 BEEP 1,-10: PAUSE 150
3040 GO TO 3550
3100 PAPER 6: INK 1: CLS
3110 PRINT ""h$(c)""ridden by
";r$(c)""DISQUALIFIED for jump
ing the""wrong fence."
3120 GO TO 3020
3500 PAPER 6: INK 1: CLS
3510 LET t(c)=INT FN t()-st
3520 PRINT TAB 12;"RESULT""""h$(
(c)""ridden by ";r$(c)""f(c);"
faults in ";t(c);" seconds"
3530 PAUSE 50: FOR a=1 TO 3: BEE
P .1,18: PAUSE 50: NEXT a
3550 CLS
3560 IF p=1 THEN GO TO 4010
3570 PRINT "TAB 7;"BEST ROUND SO
FAR""""
3580 IF f(c)>f(v) THEN GO TO 36
30
3590 IF f(c)=f(v) THEN IF t(c)>

```

```

(v) THEN GO TO 3630
4000 LET v=c
4010 IF f(v)=100 THEN PRINT TAB
41"No-one has yet finished": GO
TO 4000
4010 PRINT "Horse: ";h$(v)
4020 PRINT "Rider: ";r$(v)
4030 PRINT "Faults: ";f(v)
4040 PRINT "Time: ";t(v);" sec
4050 FOR a=1 TO 3: BEEP .6,20: N
EXT a
4060 PAUSE 400
4070 NEXT c
4070 CLS
4080 PRINT ""Do you wish to p
lay again -y/n ?"
4090 LET a$=INKEY$
4100 IF a$="y" OR a$="Y" THEN R
UN
4110 IF a$="n" OR a$="N" THEN S
TOP
4120 GO TO 4040
4130 STOP
4140 PRINT AT y,x;"A";
4150 BEEP .05,-50
4160 LET a$=INKEY$
4170 IF a$="" THEN GO TO 9010
4180 LET x1=x-(a$="z" AND x>0)+(
a$="x" AND x<31)
4190 LET y1=y-(a$="k" AND y>1)+(
a$="m" AND y<20)
4200 PRINT AT y,x;" ";
4210 LET a$=SCREEN$(y1,x1)
4220 IF a$<>" " THEN RETURN
4230 LET x=x1: LET y=y1
4240 GO TO 9000
4250 PAPER 4: INK 0: CLS
4260 FOR k=1 TO 15
4270 PRINT AT p(k,2),p(k,1);
4280 IF p(k,3)=2 THEN GO TO 936
0

```

```

9340 PRINT "BC"
9350 GO TO 9370
9360 PRINT "D";AT p(k,2)+1,p(k,1)
);"E"
9370 NEXT k
9380 RETURN
9997 DEF FN t()=FN m(FN u(),FN u
())/50
9998 DEF FN u()=(65536*PEEK 2367
4+256*PEEK 23673+PEEK 23672)
9999 DEF FN m(x,y)=(x+y+ABS (x-y
))/2

```



LEON GOODFRIEND  
Original program by Paul Bradshaw

---

## GENERAL DESCRIPTION

---

Your role in this game is of a cat guarding a large piece of cheese from hordes of hungry mice. You must prevent the mice from nibbling away at the cheese by intercepting their paths and catching them in your jaws. You gain points for how quickly you catch the mice and when all of the cheese has been nibbled the game is over and a pleasing and colourful display follows. Your score and best score so far are displayed throughout the game.

---

## DETAILED DESCRIPTION

---

Lines 10-220 Initialisation and user instructions.

230-295 Set up the screen display, draw the cheese and mark the boundaries of the screen.

300-360 Display the cat and the mouse.

380 Update the score and display the best score.

400-480 Check and update new position of mouse.

490 Check to see if all of the cheese has been eaten.

500 Check to see if the mouse has been caught in the cat's jaws.



510-520 User input to move cat.  
 530-610 Update the position of the cat.  
 8000-8080 Mouse has been captured routine.  
 9000-9470 Game over display.  
 9360-9470 Congratulatory message if new high score  
 achieved.  
 9900-9940 Another go? routine.

```

10 INK 7: PAPER 1: FLASH 0: BRIGHT 0:
OVER 0: INVERSE 0: BORDER 0: CLS
20 PRINT AT 3,13;"CHEESE"
30 PLOT 100,140: DRAW 56,0: DRAW 0,14:
DRAW -57,0: DRAW 0,-14
40 LET hs=0
50 FOR a=0 TO 15: READ b: POKE USR "a"
+a,b: NEXT a
60 DATA 66,126,66,165,129,153,231,24,4
,2,4,14,55,127,15,30
70 PRINT "'The object of the game is
to"
80 PRINT "prevent the mice b from reac
hing"
90 PRINT "the cheese in the centre of
the"
100 PRINT "screen."
110 PRINT "You control a cat a with the
"
120 PRINT "following keys:-"
130 PRINT TAB 6;"A - up      Z - down"
140 PRINT TAB 6;"N - left  M - right"
150 PRINT "Catch a mouse by interceptin
g"
160 PRINT "its path when it will run in
to"
170 PRINT "your jaws. The quicker you c
atch"
180 PRINT "a mouse the more points you"
190 PRINT "score. The game ends when al
l"
200 PRINT "the cheese is gone."
210 PRINT "'Press any key to play";

```

```

220 PAUSE 0
230 CLS
240 FOR a=10 TO 12: PRINT PAPER 6;AT a
,13;"      ": NEXT a
250 FOR a=1 TO 10
260 CIRCLE FN r(40)+108,FN r(16)+74,FN
r(2)+1
270 NEXT a
280 LET cc=18: LET sc=0
290 FOR y=0 TO 21: PRINT INK 0;AT y,0;
" ";AT y,30;" ";: NEXT y
295 FOR x=2 TO 29: PRINT INK 0;AT 0,x;
" ";AT 1,x;" ";AT 20,x;" ";AT 21,x;" ";:
NEXT x
300 LET cx=FN r(28)+1: LET cy=FN r(18)+
1: IF ATTR (cy,cx)<>15 THEN GO TO 300
310 PRINT AT cy,cx;"a"
320 LET mx=FN r(28)+1: LET my=FN r(18)+
1: IF ATTR (my,mx)<>15 THEN GO TO 320
330 IF ABS (mx-cx)^2+ABS (my-cy)^2<9 TH
IN GO TO 320
340 LET dx=SGN (RND-.5): LET dy=SGN (RN
D-.5)
350 IF ATTR (my+dy,mx+dx)<>15 THEN GO
TO 340
360 PRINT AT my,mx;"b"
370 LET md=0
380 PRINT AT 0,2: INK 7: PAPER 0;"Score
: ";sc;AT 0,16;"Best: ";hs
400 LET py=my+dy: LET px=mx+dx: LET md=
md+1
410 IF ATTR (py,px)=55 THEN GO TO 440
420 IF ATTR (py,px-1)=55 THEN LET px=p
x-1
430 IF ATTR (py,px+1)=55 THEN LET px=p
x+1
440 IF px=2 OR px=29 THEN LET dx=-dx
450 IF py=2 OR py=19 THEN LET dy=-dy
460 IF ATTR (py,px)=55 THEN LET cc=cc-
1: BEEP .2,35
470 PRINT AT my,mx;" ";AT py,px;"b"
480 LET mx=px: LET my=py

```

```

490 IF cc=0 THEN GO TO 9000
500 IF mx=cx THEN IF my=cy THEN GO TO
8000
505 IF md/2=INT (md/2) THEN BEEP .1,30
: GO TO 400
510 LET px=cx+(INKEY$="m")-(INKEY$="n")
520 LET py=cy+(INKEY$="z")-(INKEY$="a")
530 IF ATTR (py,px)=15 THEN GO TO 560
540 LET py=cy: IF ATTR (py,px)=15 THEN
GO TO 560
550 LET px=cx
560 PRINT AT cy,cx;" ";AT py,px;"a"
570 LET cy=py: LET cx=px
580 IF mx=cx THEN IF my=cy THEN GO TO
8000
610 GO TO 400
7000 DEF FN r(x)=INT (RND*x)+1
7010 DEF FN m(a,b)=a-b*INT (a/b): REM a
mod b
8000 FOR a=10 TO 0 STEP -1
8010 PRINT AT my,mx;"a"
8020 BEEP .2,2*a
8030 PRINT AT my,mx;"b"
8040 BEEP .2,2*a-1
8050 NEXT a
8060 IF md>180 THEN GO TO 320
8070 LET sc=10*INT ((sc+450-2.5*md)/10)
8080 GO TO 320
9000 PRINT AT my,mx;" ";AT cy,cx;" "
9010 INK 7: PAPER 0: FLASH 1
9020 PRINT AT 10,13;"GAME";AT 12,13;"OVE
R"
9030 PLOT 96,69
9040 DRAW 48,0: DRAW 0,16: DRAW -48,0: D
RAW 0,-16
9050 PLOT 96,84
9060 DRAW 48,0: DRAW 0,16: DRAW -48,0: D
RAW 0,-16
9070 FOR a=1 TO 750: NEXT a
9080 FLASH 0: CLS
9090 PRINT AT 11,10;"Score: ";sc
9100 PAUSE 100

```

```

110 IF sc<hs THEN PRINT AT 17,8;"High
score: ";hs: GO TO 9900
120 CLS : FOR a=21 TO 1 STEP -1
130 PRINT AT a,0;"a";
140 NEXT a
150 FOR a=0 TO 10 STEP 2
160 FOR b=a TO 31-a
170 PRINT AT a,b;"a";
180 NEXT b
190 FOR b=a TO 21-a
200 PRINT AT b,31-a;"a";
210 NEXT b
220 FOR b=31-a TO a+2 STEP -1
230 PRINT AT 21-a,b;"a";
240 NEXT b
250 FOR b=21-a TO a+2 STEP -1
260 PRINT AT b,a+2;"a";
270 NEXT b
280 NEXT a
300 FOR a=0 TO 41
310 PAPER FN m(a,8): INK FN m(a+4,8)
320 PRINT OVER 1;AT FN m(a,22),0;TAB 3
11" "
325 BEEP .04,a
330 NEXT a
340 POKE 23692,255
350 PAPER 2: INK 7
360 FOR a=1 TO 25: PRINT : NEXT a
370 PLOT 47,46: DRAW 155,0,PI/3: DRAW 0
,75,PI/3: DRAW -155,0,PI/3: DRAW 0,-75,P
I/3
380 LET a$="CONGRATULATIONS"
390 LET b$="NEW HIGH SCORE!"
400 LET a$=a$+a$: LET b$=b$+b$
410 INVERSE 1
420 FOR a=1 TO 16
430 PRINT AT 10,8;a$(a TO a+14);AT 12,8
|b$(a TO a+14)
440 BEEP .4,a*2
450 NEXT a
460 INVERSE 0
470 LET hs=sc

```



```

9900 INPUT "Another game -y/n ? ";a$
9910 IF a$<>"y" AND a$<>"n" THEN GO TO
9900
9920 IF a$="n" THEN STOP
9930 INK 7: PAPER 1
9940 GO TO 230

```

# GO-MOKU

COLLETTE WHITFIELD  
Original program by Jeff Aughton

## GENERAL DESCRIPTION

Challenge your computer to the oriental game of Go-Moku. The game is played on an 11 x 11 square board and during play you must cover one of these small squares with one of your pieces.

The object of the game is to get five adjacent pieces in a row, either horizontally, vertically or diagonally. Your pieces are coloured cyan and the computer's are yellow.

Initially the computer takes over three minutes to set up the board, then you have the option of going first or not. You take turns to place your pieces on the board, with the computer usually taking about ninety seconds to decide its move. You will be informed when someone has won and a running score is kept. The computer plays intelligently, but it is not infallible.

## DETAILED DESCRIPTION

Lines 1-205 Initial set up.

210-435 The board has been set up in memory as a table and these lines read that table.

470-545 Call up 'print board' routine.

550-660 Start the game.

670-1070 Print the playing board and get the user's input.

1250-1430 Invalid input ... try again.

1450-1880 The computer's turn.

1730-1880 Evaluate possible moves.

1890-2000 Display the move on the board.

2005-2220 Update the table in memory.

2230-2410 End of game comment.

2420-2490 Display the winner on the board.

2540-2670 Running score and another game input.

2730-2960 Instructions and rules.





```

700 PRINT AT 1,13; FLASH 1;"GOM
OKU"
710 LET mm=0
720 FOR y=3 TO 13
730 PRINT AT y,9; INK 7;y-2
740 FOR x=11 TO 21
750 PRINT AT y,x; PAPER 1-mm+3;
" "
760 LET mm=1-mm
780 NEXT x
790 NEXT y
800 PRINT AT 14,11;"ABCDEFGHJK
";
860 PRINT AT 20,0; INK 7;" Do y
ou want to start? Y/N "
870 LET A$=INKEY$
871 IF A$<>"Y" AND A$<>"N" AND
A$<>"n" AND A$<>"y" THEN GO TO
870
872 PRINT AT 20,0;"
"
890 IF A$="Y" OR A$="y" THEN L
ET M=1: RETURN
895 LET M=0: RETURN
900 REM
910 REM
950 PRINT AT 20,0;"LETTER
"
960 INPUT a$
961 IF a$<"A" OR a$>"K" THEN G
O TO 960
970 LET Y=CODE (A$)-64
980 PRINT AT 20,0;"NUMBER "
981 INPUT a$
982 IF CODE a$>57 OR CODE a$=0
THEN GO TO 981
983 LET X=INT VAL a$
985 IF X<1 OR X>11 THEN GO TO
981
1040 GO SUB 2990
1050 IF N=0 THEN GO SUB 1270
1060 IF N=0 THEN GO TO 950

```

```

1070 RETURN
1250 REM
1260 REM defprocabuse
1270 LET ERR=ERR+1
1280 GO SUB 1320+(ERR-1)*20
1290 BEEP 2,ERR
1295 PRINT AT 20,0;"
"
1300 RETURN
1320 PRINT AT 20,0;"You can't do
play there!"
1325 RETURN
1340 PRINT AT 20,0;"That square
is taken!": RETURN
1360 PRINT AT 20,0;"That's your
THIRD mistake think!": RETURN
1380 PRINT AT 20,0;"WRONG AGAIN.
YOU LOSE A TURN !"
1430 LET N=-1: LET ERR=0: RETURN

1450 PRINT AT 20,0;"LET ME THINK
....."
1470 IF FIRST<>0 THEN GO SUB 15
10: GO TO 1480
1475 GO SUB 1630
1480 PRINT AT 20,0;"I WILL PLAY
";CHR$(64+Y);" ";X
1500 PAUSE 120
1510 RETURN
1530 REM DEFPROCGUESS
1540 LET FIRST=0
1560 LET X=INT (4+RND*5): LET Y=
INT (4+RND*5)
1570 GO SUB 2990
1580 IF N=0 THEN GO TO 1560
1590 PAUSE 120
1600 RETURN
1620 REM DEFPROCTHINK
1630 IF M3S<=Y3S AND Y3S>1 THEN
LET DEF=w2: GO TO 1640
1635 LET DEF=w3
1640 IF RND*1<.9 THEN LET P1=w4

```

```

: GO TO 1650
1645 LET P1=w5
1650 LET v=-9999
1660 FOR I=LB TO UB
1670 IF C(I)=0 THEN GO SUB 1750
1680 NEXT I
1690 LET N=K: LET K=K+10
1700 LET X=1+(K-INT (K/11)*11)
1710 LET Y=INT (K/11)
1720 RETURN
1730 REM
1740 REM PROCEVAL
1750 LET T=0: LET TWOS=0: LET L=
TABLE+21*I-21
1770 LET Z=W(PEEK (L))
1780 IF Z=9 THEN GO TO 1840
1790 IF Z=4 THEN LET T=9999: GO
TO 1840
1800 IF Z=-4 THEN LET T=999
1810 IF Z=-2 THEN LET TWOS=TWOS
+1
1820 IF Z<0 THEN LET Q=DEF: GO
TO 1830
1821 LET Q=ATT
1830 LET T=T+Q*(Z+1)-P1*(Z=3)
1840 LET L=L+1
1850 IF PEEK L<>0 THEN GO TO 17
70
1860 IF TWOS>4 THEN LET T=T+w6
1870 IF T>V THEN LET V=T: LET K
=I
1880 RETURN
1890 REM
1900 REM defprocplay
1910 PRINT AT X+3-1,Y+11-1: INK
6+M-1: FLASH 1;" ";
1911 FOR I=1 TO 10: BEEP .1,2: N
EXT I
1980 PRINT AT X+3-1,Y+11-1: INK
6+M-1: FLASH 0;"A";
1990 GO SUB 2020
2000 RETURN

```

```

2010 REM DEFPROCUDATE
2020 LET P=2*M-1: LET C(N)=P
2040 LET K=TABLE+21*N-21
2050 LET Q=PEEK K
2070 LET V=W(Q)
2080 IF V=9 THEN GO TO 2150
2090 IF V*P<0 THEN LET W(Q)=9:
LET WIN=WIN-1: GO TO 2100
2095 LET W(Q)=V+P
2100 IF V=-3 THEN LET Y3S=Y3S-1
2110 IF V=3 THEN LET M3S=M3S-1
2120 IF V=-2 AND P<0 THEN LET Y
3S=Y3S+1
2130 IF V=2 AND P>0 THEN LET M3
S=M3S+1
2140 IF ABS (W(Q))=5 THEN LET M
=INT ((P+5)/2): LET WIN=Q
2150 LET K=K+1: LET Q=PEEK K
2160 IF Q<>0 THEN GO TO 2070
2170 IF WIN=0 THEN LET M=4
2180 LET N=N-23
2190 IF N>0 AND N<LB THEN LET L
B=N
2200 LET N=N+46
2210 IF N<122 AND N>UB THEN LET
UB=N
2220 RETURN
2240 REM DEFPROCCOMMENTS
2250 IF M<4 THEN GO SUB 2440
2255 CLS
2256 LET A$=""
2260 GO SUB 2310+(M-2)*20
2270 PRINT AT T,0;A$
2280 PAUSE 120
2290 RETURN
2310 LET YWIN=YWIN+1: LET A$="YO
U WIN!!!"
2320 LET B$="WELL DONE!"
2325 LET T=5: RETURN
2330 LET MWIN=MWIN+1: LET A$="I'
ve won!!!"
2335 LET B$="I Really enjoyed th

```



```

at!!"
2340 LET T=4: RETURN
2350 LET DRAW=DRAW+1: LET A$="It
's a draw!!"
2380 LET B$="No won could have w
on that!"
2410 LET T=3: RETURN
2430 REM DEFROCSHOWIN
2440 PRINT AT X+3-1,Y+11-1; FLAS
H 1;"W";
2460 FOR I=1 TO 5: BEEP 1,1: NEX
T I
2490 RETURN
2540 REM
2560 REM DEFPROCGOVER
2570 PRINT B$
2580 PRINT "THE SCORE SO FAR"
2590 PRINT "YOU   =";YWIN
2600 PRINT "ME     =";MWIN
2610 PRINT "DRAWN =";DRAW
2630 PRINT "Do you want to play
again ? Y/N"
2640 LET A$=INKEY$
2645 IF A$<>"N" AND A$<>"Y" THEN
GO TO 2640
2660 IF A$="N" THEN LET DONE=1
2670 RETURN
2740 REM DEFPROCRRULES
2750 BORDER 0: INK 7: PAPER 0: C
LS
2755 PRINT AT 1,10;"Gomoku": PRI
NT
2760 PRINT "This game is similar
to noughts-"
2770 PRINT "and-crosses,the obje
ct being to"
2780 PRINT "create FIVE OR MORE
pieces"
2790 PRINT "either horizontally,
vertically "
2800 PRINT "or diagonally."
2810 PRINT "My pieces are "; INK

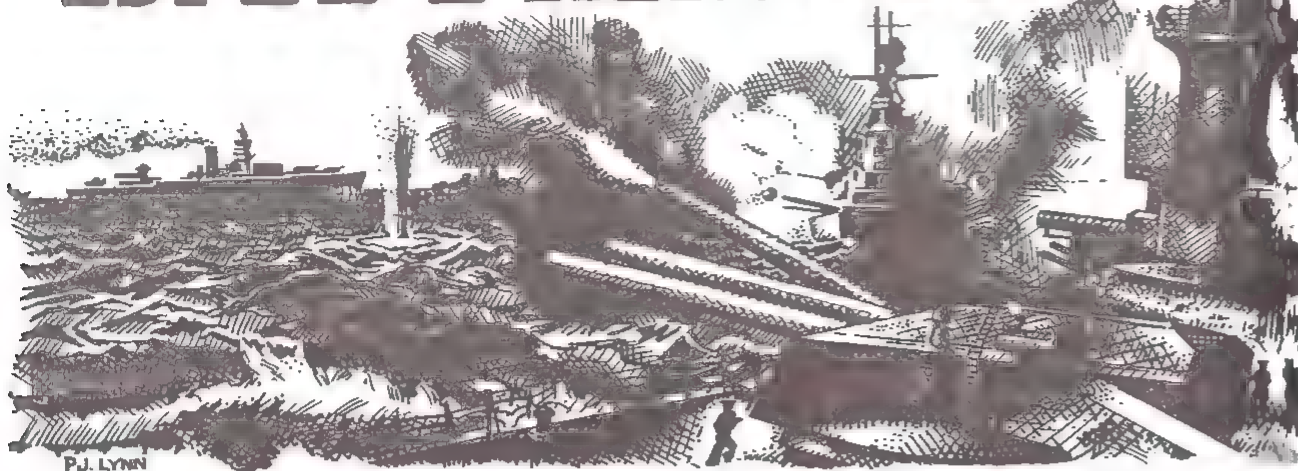
```

```

6;"YELLOW"; INK 7;" and yours"
2820 PRINT "are "; INK 5;"CYAN";
INK 7
2830 PRINT
2840 PRINT "Best of luck,because
youll need it."
2850 PRINT : PRINT "PLEASE GIVE
ME THREE MINUTES TO          S
ET UP!"
2960 RETURN
2990 LET N=11*Y-11+X
3000 IF C(N)<>0 THEN LET N=0
3010 RETURN
3050 DATA 10,-61,-20,220,55,135
3060 DATA 16,56,124,124,124,124,
56,16
3070 DATA 1,11,1,1,4,1,11,11,11,
-76,1,7,12,1,4,5,7,10,1,4

```

# BATTLESHIPS



ANDREW CRESSWELL  
Original program by J. Edyvane

## GENERAL DESCRIPTION

The object of Battleships is to sink the computer's fleet before it sinks yours. First you must place your six ships on the grid. To do this follow the prompts and enter the grid references: letter, number then (ENTER). Each ship must form a horizontal, vertical or diagonal line. You will see your fleet appearing on the left-hand grid as you input this information.

You must now attempt to locate and sink the computer's entire fleet by firing three shell salvos at the right-hand grid. Again type a letter, followed by a number, followed by (ENTER). The result of your action will be displayed graphically on the computer's grid, and in words under your grid. Now it is the computer's turn to fire at your grid.

The game continues in this fashion until either of the fleets is totally destroyed, and the winner is announced.

## DETAILED DESCRIPTION

Lines 20-80 Initialise and read data for ship types.  
160-190 Player's input and validation.

200-275 Work out where to fire the computer's three-shell salvo.

300-890 Main program which first draws the two grids then sets up the computer's ships and asks the player to do likewise.

670-890 Fire routine and resultant actions.

900-999 This routine is called when your shot misses.

1000-9992 Sets up the user-defined characters.

```

1 REM *****
2 REM ***** Battleships *****
3 REM ***** and *****
4 REM ***** Cruisers *****
5 REM *****
6 REM Original ZX81 program
7 REM by J.C. Edyvane
8 REM *****
9 REM Spectrum conversion
10 REM by A. Cresswell
11 REM *****
15 GO SUB 1000
20 REM initialise
21 BORDER 1: PAPER 5: CLS : INK 7
22 POKE 23658,8: POKE 23609,50
25 LET h=0: LET i=0: LET j=0
30 DIM a(2,10,10): DIM n(2,6): DIM n$(
6,11)
60 RANDOMIZE
65 RESTORE 70: FOR i=1 TO 6: READ x$:
LET n$(i)=x$: NEXT i
70 DATA "Battleship","Cruiser","Destro
yer 1","Destroyer 2","Submarine 1","Subm
arine 2"
75 FOR s=1 TO 2: RESTORE 77: FOR i=1 T
O 6: READ x: LET n(s,i)=x: NEXT i: NEXT
n
77 DATA 5,4,3,3,2,2
80 GO TO 300
160 REM get player's input
165 INPUT a$: IF a$="" THEN GO TO 165
170 LET x=(CODE a$(1))-64: IF x>32 AND

```



```

x<43 THEN LET x=x-32
175 IF x<1 OR x>10 THEN GO TO 165
180 LET a$=a$(2 TO ): IF CODE a$<48 OR
CODE a$>57 THEN GO TO 165
185 LET y=INT (VAL a$)+1: IF y<1 OR y>1
0 THEN GO TO 165
187 IF a(2,x,y)<0 THEN GO TO 165
190 RETURN
200 REM computer's fire routine
202 IF h<>0 AND j<>0 THEN GO TO 222
205 IF h<>0 THEN GO TO 230
210 LET x=INT (RND*10)+1: LET y=INT (RN
D*10)+1: IF a(1,x,y)<0 THEN GO TO 210
220 RETURN
222 LET p=h: LET q=i
230 FOR d=-1 TO 1: FOR u=-1 TO 1: LET x
=u+p: IF x<1 OR x>10 THEN GO TO 263
235 LET y=d+q: IF y<1 OR y>10 THEN GO
TO 265
261 IF a(1,x,y)<0 THEN GO TO 263
262 RETURN
263 NEXT u
265 NEXT d
270 LET p=h1: LET q=i1: GO TO 230
275 GO TO 210
300 REM main program
301 PAPER 7: INK 0
305 PRINT PAPER 5;" "; PAPER 7;" Your
fleet "; PAPER 5;" "; PAPER 7;"Enemy
fleet ": PRINT
310 LET b$=" ABCDEFGHIJ ": PRINT PAPER
5;" "; PAPER 7;b$; PAPER 5;" "; PAP
ER 7;b$
320 FOR s=3 TO 12: RESTORE 330: FOR i=1
TO 4: READ x: PRINT AT s,x; INVERSE 1;C
HR$ (60-s): NEXT i: NEXT s
330 DATA 2,13,18,29
340 PRINT PAPER 5;" "; PAPER 7;b$; PA
PER 5;" "; PAPER 7;b$: PRINT AT 15,2;
INK 2; FLASH 1;"I am positioning my fle
et".
400 FOR s=1 TO 6

```

```

410 LET d=INT (RND*3)-1: LET u=INT (RND
*3)-1: IF d=0 AND u=0 THEN GO TO 410
420 LET p=INT (RND*10)+1: LET q=INT (RN
D*10)+1
430 LET m=n(1,s)-1: LET t1=p+d*m: LET t
2=q+u*m
440 IF t1>10 OR t1<1 OR t2>10 OR t2<1 T
HEN GO TO 410
450 FOR l=0 TO m: IF a(2,p+d*l,q+u*l)<>
0 THEN GO TO 410
460 NEXT l
470 FOR l=0 TO m: LET a(2,p+d*l,q+u*l)=
a1 NEXT l
500 NEXT s
530 PRINT AT 15,2;"Enter your ships now
"
570 FOR s=1 TO 6: PRINT AT 17,2;n$(s)
590 FOR l=1 TO n(2,s): PRINT AT 19,2;"S
quare ";l: GO SUB 160
595 IF a(1,x,y)<>0 THEN GO SUB 160: GO
TO 595
600 LET a(1,x,y)=s: PRINT AT 13-y,x+2;n
$(s,1): NEXT l: NEXT s
630 PRINT AT 15,2; PAPER 5;"
"
640 PRINT AT 17,2; PAPER 5;"
"
650 PRINT AT 19,2; PAPER 5;"
"
660 LET k=2
670 IF k=1 THEN PRINT AT 15,2;"My go
"
680 IF k=2 THEN PRINT AT 15,2;"Your go
"
690 FOR g=1 TO 3
700 FOR s=1 TO 20: NEXT s
710 PRINT AT 17,2; PAPER 5;"
"
720 PRINT AT 16,2;"Fire ";g
730 IF k=1 THEN GO SUB 200
740 IF k=2 THEN GO SUB 160
750 LET z=a(k,x,y): LET a(k,x,y)=-1: IF

```

```

z=0 THEN GO TO 900
760 PRINT AT 13-y,x+(k*16)-14; FLASH 1;
PAPER 2; INK 6;"b"
770 PRINT AT 17,2;"BANG"; PAPER 5;"
      ": BEEP 1,1
780 IF k=1 THEN IF h=0 THEN LET h1=x:
LET i1=y
785 IF k=1 THEN LET h=x: LET i=y: LET
j=1
790 LET n(k,z)=n(k,z)-1
800 IF n(k,z)<>0 THEN GO TO 875
805 PRINT AT 17,2;n$(z);" sunk"
810 IF k=1 THEN LET h=0
815 FOR s=1 TO 50: NEXT s
820 FOR s=1 TO 6: IF n(k,s)>0 THEN GO
TO 875
825 NEXT s
830 IF k=2 THEN PRINT AT 19,2; INK 6;
PAPER 2; FLASH 1;"You ";
835 IF k=1 THEN PRINT AT 19,2; INK 6;
PAPER 2; FLASH 1;" I ";
840 PRINT INK 6; PAPER 2; FLASH 1;"win
."
850 FOR k=1 TO 2: FOR x=1 TO 10: FOR y=
1 TO 10
855 IF a(k,x,y)=0 THEN PRINT AT 13-y,x
+(k*16)-14;"a": BEEP .1,1
860 IF a(k,x,y)>0 THEN PRINT AT 13-y,x
+(k*16)-14; FLASH 1; PAPER 2; INK 6;"b":
BEEP 1,1
865 NEXT y: NEXT x: NEXT k
870 GO TO 9990
875 IF k=1 THEN PAUSE 50
880 NEXT g
890 LET k=1+ABS (k-2): GO TO 670
900 REM      misfire routine
905 PRINT AT 13-y,x+(k*16)-14;"a": BEEP
.1,1
910 PRINT AT 17,2;"SPLASH"; PAPER 5;"
      "
915 IF k=1 THEN LET j=0
920 GO TO 875

```

```

999 GO TO 9990
1000 REM sets up graphic char
1010 RESTORE 1020: FOR n=USR "a" TO USR
"b"+7: READ nn: POKE n,nn: NEXT n: RETURN
1020 DATA 85,170,85,170,85,170,85,170
1030 DATA 153,90,90,255,255,90,90,153
1040 PRINT AT 21,0;" Press any key for
another go "
1050 IF INKEY#="" THEN GO TO 9991
1060 GO TO 20

```



# DUCKSHOOT

JEREMY HANNETT  
Original program by B. Philips

## GENERAL DESCRIPTION

Armed with your double-barrelled shotgun and with your trusty gundog by your side you stomp through the marshland, trying to shoot down as many ducks as you can. The ducks will not fly when the sun is out, so you can gain extra points by shooting at them. Bonus points are gained with every five ducks shot, as they are taken to market. The duck shooting season is very short, so you must shoot quickly. Your score and the time remaining are displayed at the top of the screen.

## DETAILED DESCRIPTION

Lines 1-50 Title page is displayed and the user-defined graphics subroutine is called.  
90-130 Print marshland and either duck or sun.  
200-500 Main program loop.  
600-610 Duck escaped.  
620-730 Duck hit and retrieved by the dog.  
800-840 Display the shot.  
850-900 Hit something.  
2000-2035 Print and move the hunter.  
2100-2110 Fire at target.  
6000-6060 Print score and ask for another go.  
8000-8220 Instructions.  
9000-9020 Set up the user-defined graphics characters.  
9100-9130 Set time and read the clock.

```
1  REM  *** Duck shoot
2  PAPER 7: BORDER 7
3  CLS : FOR n=7 TO 0 STEP -1: PRINT
INK n; INVERSE 1;"      D U C K  -  S H O
4  T .      ": NEXT n
5  FOR n=0 TO 7: PRINT INK n; INVERSE
1;"      D U C K  -  S H O O T .      ": N
EXT n
6  LET e$=""
7
8  10 PRINT AT 21,0; FLASH 1;"Do you want
instructions ? (Y/N)"
9  20 LET a$=INKEY$: IF a$="" THEN GO TO
20
10 30 IF a$="y" OR a$="Y" THEN GO SUB 80
11
12 40 GO SUB 9000
13 50 GO SUB 9100
14 90 PAPER 5: BORDER 1
15 100 CLS : PRINT AT 19,0; INK 4;"aaaaaaa
16  aaaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbb
17  bbbbbbbbbbbbbbbbbbbcccccccccccccccccccccc
18  cccccccc";
19 105 LET ss=0: LET ducks=0: LET shotx=0:
20  LET shoty=0: LET sun=0: LET escaped=0:
21  LET score=0
22 110 LET position=1: LET char=1: GO SUB
23  2000
24 120 LET ducky=INT (RND*15)+2: LET duckx
25  =1
26 130 IF RND>.8 THEN LET sun=1: LET sunx
27  =INT (RND*240)+5: LET suny=INT (RND*100)
28  +60: FOR n=1 TO 5: CIRCLE OVER 1; INK 6
29  1sunx,suny,n: NEXT n
30 200 REM *** Main loop
31 210 GO SUB 9110: PRINT AT 0,0; INVERSE
32  11e$;AT 0,0;"Time Left =";INT (300-time)
33  1AT 0,14;" Secs. Score =";Score
34 220 IF time>=299 THEN GO TO 6000
35 230 LET a$=INKEY$: IF a$="j" THEN LET
36  position=position+1: LET char=2: GO SUB
37  2000: BEEP .1,0: LET char=3: GO SUB 2000
```

```

240 IF a$="g" THEN LET position=position-1: LET char=4: GO SUB 2000: BEEP .1,0
: LET char=5: GO SUB 2000
250 IF a$="" THEN LET char=1: GO SUB 2000
260 IF a$="h" THEN GO SUB 900: LET char=1: GO SUB 2000: LET shoty=35: LET shotx=(position*8)+7
265 IF shoty<>0 THEN GO TO 800
266 IF sun=1 THEN GO TO 210
270 PRINT AT ducky,duckx;" no";: FOR d=1 TO 5: BEEP .01,d: NEXT d: PRINT AT ducky,duckx+1;"pq"
280 LET duckx=duckx+1: IF duckx>=28 THEN PRINT AT ducky,duckx;" ": GO TO 600
500 GO TO 210
600 REM *** Duck escaped.
610 LET escaped=escaped+1: GO TO 120
620 REM *** Killed duck!
625 FOR n=25 TO -10 STEP -5: BEEP .01,n: NEXT n
630 FOR n=ducky TO 18: PRINT AT n,duckx;" ": LET duckx=duckx+1: IF duckx>=28 THEN LET duckx=0
640 PRINT AT n+1,duckx; INK 2;"pq": BEEP .05,(10-n): NEXT n
650 FOR n=0 TO duckx: PRINT AT 19,n; INK 4;"a"; INK 0;"u": BEEP .05,n+12: PRINT AT 19,n+1; INK 0;"s": BEEP .05,n: GO SUB 2000: NEXT n: IF duckx=position THEN GO TO 700
651 IF duckx>position THEN GO TO 670
660 FOR n=duckx TO position-1: PRINT AT 19,n; INK 4;"a"; INK 0;"s": BEEP .1,0: PRINT AT 19,n+1; INK 0;"u": BEEP .1,0: GO SUB 2000: NEXT n: PRINT AT 19,position-1; INK 4;"a": GO TO 700
670 FOR n=duckx TO position+2 STEP -1: PRINT AT 19,n; INK 0;"r"; INK 4;"a": BEEP .1,0: PRINT AT 19,n; INK 0;"t": BEEP .1,0: GO SUB 2000: NEXT n: PRINT AT 19,position+2; INK 4;"a": GO TO 700

```

```

700 LET score=score+1: LET shot=0: LET ducks=ducks+1
701 LET x=score
702 IF x>5 THEN LET x=x-5: GO TO 702
705 LET x=x-1: PRINT AT 21,x*2;"m1"
710 IF INT (score/5)=score/5 THEN GO SUB 730
720 GO TO 120
730 FOR n=1 TO 5: FOR f=1 TO 10 STEP 2: BEEP .02,n+f: PRINT AT 21,10-(n*2);"cc"
740 RETURN
800 REM *** shot on screen
805 LET count=0
810 GO SUB 900: LET shoty=shoty+4: PLOT shotx,shoty: DRAW 0,2
820 IF POINT (shotx,shoty+3)=1 OR POINT (shotx,shoty+4)=1 OR POINT (shotx,shoty+5)=1 THEN GO TO 850
830 IF shoty>150 THEN GO SUB 900: LET shoty=0: GO TO 266
835 LET count=count+1: IF count<5 THEN GO TO 810
840 GO TO 266
850 REM *** Hit Something
860 GO SUB 900: LET shoty=0: IF sun=0 THEN GO TO 620
865 FOR n=5 TO 1 STEP -1: CIRCLE INVERSE 1; INK 6;sunx,suny,n: BEEP .01,n*3: NEXT n
870 LET ss=ss+1: LET sun=0: LET score=score+25: GO TO 120
900 PLOT INVERSE 1;shotx,shoty: DRAW INVERSE 1;0,2: RETURN
2000 REM *** Print Hunter
2001 IF position=0 THEN PRINT AT 18,0;" ";AT 19,0; INK 4;"aaa": LET position=10: GO TO 2010
2002 IF position=31 THEN PRINT AT 18,28;" ";AT 19,28; INK 4;"aaa": LET position=1: GO TO 2010
2005 PRINT AT 18,position-1;" ";AT 19,

```

```

position-1; INK 4;"aaa"
2010 IF char=1 THEN PRINT AT 18,position; INK 0;"d";AT 19,position; INK 0;"e";: RETURN
2020 IF char=2 THEN PRINT AT 18,position; INK 0;"f";AT 19,position; INK 0;"g";: RETURN
2025 IF char=3 THEN PRINT AT 18,position; INK 0;"f";AT 19,position; INK 0;"h";: RETURN
2030 IF char=4 THEN PRINT AT 18,position; INK 0;"i";AT 19,position; INK 0;"j";: RETURN
2035 IF char=5 THEN PRINT AT 18,position; INK 0;"i";AT 19,position; INK 0;"k";: RETURN
2100 REM *** fire at target
2110 LET shot=17: LET shotpos=position: RETURN
6000 CLS : FOR n=1 TO 3: PRINT FLASH 1; BRIGHT 1;e$: NEXT n
6010 PRINT AT 1,10; FLASH 1; INVERSE 1;"You Beast!!!"
6020 PRINT AT 7,5;"You have just killed ";ducks: PRINT " poor little ducks !!": PRINT : PRINT : IF escaped>0 THEN PRINT " Fortunately, ";escaped;" got away.": PRINT : PRINT : PRINT " You put out the sun ";ss;" times,"
6030 PRINT " and scored ";score;" points."
6040 PRINT : PRINT : PRINT FLASH 1;" Press any key to play again. "
6050 LET a$=INKEY$: IF a$="" THEN GO TO 6050
6060 RUN
3000 REM *** Instructions
8010 CLS : PRINT AT 0,0; BRIGHT 1; FLASH 1;" D U C K - S H O O T "
8020 PRINT AT 2,0;"Instructions :-"
8030 PRINT "This is a wicked game of skill at Duck Shooting."

```

```

8040 PRINT "You are stomping through the marsh-land, waiting for poor unsuspecting ducks to fly over."
8050 PRINT "Every duck you shoot is laid out on the ground around you, so you may count your spoils. When you have shot 5 ducks, they are taken off to market, and you are given 15 Bonus points."
8060 PRINT "When the sun comes out, the ducks do not fly. You may then take a pot shot at the sun to gain extra points."
8070 GO SUB 8200
8080 PRINT AT 2,0;"Every 5 Ducks you shoot gives a Bonus of 15 Points."
8090 PRINT "Every time you hit the sun, you get an extra 25 Points.": PRINT : PRINT "Controls are as follow :-"
9100 PRINT : PRINT " 'G' moves you LEFT. 'J' moves you RIGHT. 'H' fires your gun."
9110 PRINT AT 15,0; INVERSE 1;" YOU R TIME IS LIMITED SO SHOOT CAREFULLY!! "
9200 PRINT AT 21,0; FLASH 1;" Press any key to continue. "
9210 LET a$=INKEY$: IF a$="" THEN GO TO 9210
9215 FOR f=1 TO 21: PRINT AT f,0;e$;: NEXT f
9220 RETURN
9000 REM *** set up user graphics
9001 IF PEEK USR "a"=16 AND PEEK USR "b"=0 THEN RETURN
9005 RESTORE 9010: FOR n=USR "a" TO USR "a"+7: READ a: POKE n,a: NEXT n: RETURN
9010 DATA 16,8,138,76,168,106,28,255,0,20,193,38,24,193,38,153
9011 DATA 0,255,0,255,0,0,255,0,1,49,49,1,127,251,237,153
9012 DATA 120,120,72,72,72,72,206,0,40,48,0,120,252,250,186
9013 DATA 56,62,34,34,35,32,32,56,120,56

```



```
,40,40,232,136,136,14
9014 DATA 0,12,12,0,30,63,93,93,28,124,6
8,68,196,4,4,28
9015 DATA 30,28,20,20,23,17,17,112,0,24,
62,224,240,240,224,192
9016 DATA 64,64,96,63,63,31,15,3,1,3,3,2
7,252,15,0,0
9017 DATA 128,128,128,220,127,192,0,0,0,
0,7,30,253,11,3,6
9018 DATA 0,0,128,92,255,128,0,0,98,225,
63,62,34,34,68,0
9019 DATA 70,135,252,124,68,68,34,0
9020 DATA 97,225,63,62,66,66,33,0,134,13
5,252,124,66,66,132,0
9100 REM *** set time to 0
9101 LET frames=23672: POKE frames,0: PO
KE frames+1,0: POKE frames+2,0: RETURN
9110 REM *** read clock
9120 LET time=(65536*PEEK (frames+2))+(2
56*PEEK (frames+1))+PEEK frames
9130 LET time=time/50: RETURN
```



STEVE BROCKBANK  
Original program by Simon Williams

## GENERAL DESCRIPTION

Use your computer to place the past into more meaningful perspective, understand the present and reveal the alternatives which exist in the future with this 'Tarot' listing.

The Tarot pack is the forerunner of modern-day playing cards. The pack consists of 78 cards divided into 22 major arcana and 56 minor arcana cards. The 4 suits of 14 cards are named Wands, Cups, Swords and Pentacles. Each suit contains 10 numbered cards and 4 face cards: King, Queen, Knight and Page. If a card is upside down or reversed then the meaning of the card is weakened, delayed or even reversed, but the program takes this into account.

Upon running you will be asked some questions so that a 'significator' card can be assigned to you. Then you must concentrate on a matter of concern and the computer reads from the 10 card or Celtic Cross 'lay'. You can treat this divination light-heartedly or give it some deeper significance.

## DETAILED DESCRIPTION

Lines 10-25 Set variables and dimension the arrays.

30-240 Information, instructions and the assignment of your 'significator' card.

250-370 Reading from 10 card Celtic Cross 'lay'.

380-515 Card and message selection and print.

529-550 Opening title page.

690-710 Data for card names.

720-1150 Data for divinatory meanings for each of the cards.

1500-1520 User prompt.

2000-3099 Set up user-defined graphics characters subroutine.

```

5 POKE 23658,0: REM **lower c
ase**
10 RANDOMIZE : GO SUB 530
15 LET n1=1: LET r1=1: LET r2=
1: LET r=1: LET cn=0: LET bor=3
25 DIM c(11): DIM c$(6)
30 REM *** Instructions ***
40 BORDER 1: INK 0: PAPER 6: C
LS
50 PRINT "The Tarot is an anci
ent method"
52 PRINT "of fortune telling w
hich makes"
54 PRINT "use of a pack of 78
cards."
56 PRINT : PRINT " These are
divided into two"
58 PRINT "sections. The major
and minor "
60 PRINT "'arcanas'.": PRINT :
PRINT "The major arcana is a se
t of 22"
62 PRINT "pictorial cards depi
cting people and objects."
66 PRINT : PRINT "The minor ar

```

cana is similar to "

```

68 PRINT "an ordinary pack of
playing"

```

```

70 PRINT "cards with an extra
card in each"

```

```

72 PRINT "suit, the 'page'. Th
e four suits"

```

```

74 PRINT "are named wands, cup
s, swords &"

```

```

76 PRINT "pentacles.": GO SUB
1500: CLS

```

```

78 PRINT "The major and minor
arcana are shuffled together an
d dealt out,"

```

```

80 PRINT "face down, in any of
several"

```

```

82 PRINT "arrangements known a
s 'lays'."

```

```

84 PRINT : PRINT "The cards ar
e then revealed, one at a time, a
nd read according to"

```

```

90 PRINT "their individual mea
nings and their positions in r
elation to"

```

```

94 PRINT "the other cards.": P
RINT " I shall read from the 10
card "

```

```

96 PRINT " or 'Celtic Cross' l
ay."

```

```

100 PRINT : PRINT "Before we st
art, I need to know"

```

```

102 PRINT "some things about yo
u, in order"

```

```

104 PRINT "to assign you one of
the court "

```

```

106 PRINT "cards as your 'signi
ficator'."

```

```

107 GO SUB 1500: CLS

```

```

108 PRINT : INK 1: PRINT "Are y
ou male or female?"

```

```

109 PRINT "(Answer m or f)"

```

```

110 LET s$=INKEY$: IF s$<>"m"

```

```

AND s$<>"f" THEN GO TO 110
112 PRINT : PRINT "Are you over
40?"
114 PRINT "(Answer y or n)"
116 LET a$=INKEY$: IF a$<>"y"
AND a$<>"n" THEN GO TO 116
120 CLS
130 PRINT "What colour is your
hair?"
132 PRINT "A = very fair"
134 PRINT "B = gray"
136 PRINT "C = light brown"
138 PRINT "D = dark brown"
140 PRINT "E = black"
142 LET h$=INKEY$: IF h$<>"a"
AND h$<>"b" AND h$<>"c" AND h$<>
"d" AND h$<>"e" THEN GO TO 142
150 LET a$=s$+a$
152 IF a$="my" THEN LET a$="KI
NG": LET a=14: GO TO 180
160 IF a$="fy" THEN LET a$="QU
EEN": LET a=13: GO TO 180
170 IF a$="mn" THEN LET a$="KN
IGHT": LET a=12: GO TO 180
172 LET a$="PAGE": LET a=11
180 LET a$=a$+" of "
182 IF h$="a" OR h$="b" THEN L
ET h$="WANDS": LET h=64: GO TO 2
10
190 IF h$="c" THEN LET h$="CUP
S": LET h=50: GO TO 210
200 IF h$="d" THEN LET h$="SWO
RDS": LET h=36: GO TO 210
202 LET h$="PENTACLES": LET h=2
2
210 LET c(1)=a+h
212 CLS : PRINT "Thank you. You
r signicator is"
214 INVERSE 1: PRINT AT 2,8;a$+
h$
216 INVERSE 0
218 PRINT : PRINT "This card is

```

```

placed face up on "
220 PRINT "the table and the fi
rst 2 cards"
222 PRINT "of the lay are place
d on top of"
224 PRINT "it as they are dealt
"
230 PRINT : PRINT "Please conce
ntrate on any matter"
232 PRINT "of concern before pr
essing SPACE"
234 PRINT "to begin the reading
"
240 IF INKEY$<>" " THEN LET r=
RND: GO TO 240
250 REM *** reading ***
260 CLS
262 PRINT "The first card is la
id on top of"
264 PRINT "the signicator and
shows the"
266 PRINT "general surroundings
of your "
268 PRINT "enquiry. The card I
turn up is": GO SUB 380
270 CLS : PRINT "The second car
d is placed across"
272 PRINT "the first and indica
tes any "
274 PRINT "current problems. Fa
vourable "
276 PRINT "cards mean small pro
blems."
278 PRINT "I draw ": GO SUB 38
0
280 CLS : PRINT "The third card
, which indicates"
282 PRINT "your aims and ideals
, is placed"
284 PRINT "above the significat
or. In this"
286 PRINT "case the card is:":

```



```

GO SUB 380
290 CLS : PRINT "The fourth card, placed below "
292 PRINT "the significator, shows the "
294 PRINT "background to the present"
296 PRINT "situation - what led up to it."
298 PRINT "I turn up:": GO SUB 380
300 CLS : PRINT "Left of the significator, the"
302 PRINT "fifth card shows the immediate "
304 PRINT "past - what is 'behind' you. "
306 PRINT "This card is:": GO SUB 380
310 CLS : PRINT "The sixth card, laid to the "
312 PRINT "right of the significator, shows"
314 PRINT "the immediate future - what is "
316 PRINT "'before' you. The card drawn is:": GO SUB 380
320 CLS : PRINT "The seventh card, placed to the"
322 PRINT "right of the sixth, shows your"
324 PRINT "attitude to the matter. If a "
326 PRINT "court card is drawn, you are "
328 PRINT "showing some of the features of"
329 PRINT "that character. I turn up:": GO SUB 380
330 CLS : PRINT "The eighth card, which indicates"
332 PRINT "the environment in w

```

```

high you are"
334 PRINT "working, is laid above the "
336 PRINT "seventh. I draw:": GO SUB 380
340 CLS : PRINT "The ninth card shows your hopes"
342 PRINT "and fears and is placed above "
344 PRINT "the eighth. This card is:": GO SUB 380
350 CLS : PRINT "The last card, the tenth, shows"
352 PRINT "the final outcome of the matter"
354 PRINT "- what will be - and is laid "
356 PRINT "above the ninth. It is:": GO SUB 380
360 CLS : PRINT AT 9,0;"This completes the reading."
362 PRINT AT 11,0;"I wish you good fortune."
364 PAUSE 500
370 RESTORE : GO SUB 530: RUN
380 REM :# card/message selection and print #*
390 LET r=INT (RND*(156)+1): IF r>78 THEN LET n$=" REVERSED": LET r1=r-78
395 IF r<=78 THEN LET n$="": LET r1=r
400 IF r1>64 THEN LET r2=r1-64: LET n$=" of WANDS"+n$: GO TO 450
410 IF r1>50 THEN LET r2=r1-50: LET n$=" of CUPS"+n$: GO TO 450
420 IF r1>36 THEN LET r2=r1-36: LET n$=" of SWORDS"+n$: GO TO 450
430 IF r1>22 THEN LET r2=r1-22

```

```

: LET n$=" of PENTACLES"+n$: GO
TO 450
440 RESTORE 690: FOR n=1 TO r1:
READ m$: NEXT n: LET n$=m$+n$:
LET j$="MAJOR": GO TO 460
450 RESTORE 710: FOR n=1 TO r2:
READ c$: NEXT n: LET n$="The "+
c$+n$: LET j$="MINOR"
460 IF n1>=12 THEN LET n1=1: G
O TO 470
462 IF r=c(n1) THEN LET n1=1:
GO TO 380
465 LET n1=n1+1: GO TO 460
470 LET c(cn+1)=r: RESTORE 720:
FOR n=1 TO r: READ t$: NEXT n
475 PRINT : PRINT n$
480 PRINT : PRINT "This card is
of the ";j$;" Arcana"
482 PRINT "and implies:": PRINT
: PRINT ;t$
490 LET cn=cn+1: GO SUB 2000
500 GO SUB 1500
510 LET bor=bor+1: IF bor>7 THE
N LET bor=0
515 BORDER bor: RETURN
530 BORDER 0: PAPER 2: INK 7: C
LS
540 INVERSE 1: PRINT AT 8,5;"*
T * A * R * O * T *": INVERSE 0
550 GO SUB 3000: LET cn=10: GO
SUB 2000: PAUSE 200: RETURN
690 DATA "The MAGICIAN","The HI
GH PRIESTESS","The EMPRESS","The
EMPEROR","The HEIROPHANT","The
LOVERS","The CHARIOT","JUSTICE",
"The HERMIT","The WHEEL of FORTU
NE"
700 DATA "STRENGTH","The HANGE
D MAN","DEATH","TEMPERANCE","The
DEVIL","The TOWER","The STAR","
The MOON","The SUN","JUDGEMENT",
"The WORLD","The FOOL"

```

```

710 DATA "ACE","2","3","4","5",
"6","7","8","9","10","PAGE","KNI
GHT","QUEEN","KING"
720 DATA " Either the need for
or the skills of diplomacy."
,"Mystery & the future of a woma
n with psychic powers.", " Fruit
fulness in all its associa
tions. Initiative."
725 DATA "Stability in life. Th
e aid of a powerful person."
730 DATA " Associations with
others and particularly marriag
e.", "All forms of love and attra
ction to beautiful things"
732 DATA "Triumph only thru' pe
rseverance and doggedness","The
just outcome of problems."
740 DATA "Corruption. Others ma
y try to betray you.", "The for
ces of luck and therefore succes
s."
742 DATA "A time for taking cou
rage and acting energetically.
","Intuition. Your own or advice
from one who has."
750 DATA "The end of a cycle o
f events. Change and renewal."
,"A need for economy and good
management."
752 DATA "The effect of extra e
ffort. Possibly violence","A
diversity and distress. Misery c
aused by conflict."
760 DATA "Loss by theft of mate
rial or spiritual things.", "A
warning of hidden enemies who w
ill deceive."
762 DATA "Material happiness. C
ontentment through good marriage
."
770 DATA "Change of position. R

```

enewal and rebirth.", "Assured success via a voyage or change of place.", "A time to consider your career. Consolidation."

780 DATA "Perfect contentment. Also financial reward.", "A time for gaiety & recreation. Perhaps good news."

782 DATA "Craftsmanship and the need to use practical gifts.", "Gift or legacy. The acquisition of new possessions."

790 DATA "Material difficulties but love is well looked on.", "Gratification through gifts or other pleasures."

792 DATA "Ingenuity in business and money matters.", "A creative period in craft or business."

800 DATA "Prudence. Safety and accomplish-ment from this.", "Gain in monetary or family matters."

802 DATA "A time for scholarship or reflection. Look to your affairs", "The effect of some one with your affairs at heart."

810 DATA "The effects of an opulent though generous woman. Security.", "Intellect and particularly mathematics. Look to business."

820 DATA "Great force in love or hate. Triumph through this.", "Conformity and the stability this may bring."

822 DATA "A feeling of loss of someone or something.", "The need for solitude. A time for contemplation."

830 DATA "Loss or destruction of something close to you.", "A joy

ney which may include travel by or over water."

832 DATA "Annoyance at the failure of a plan through disagreement.", "Bad news of an illness or other crisis."

840 DATA "Deep disappointment possibly through death.", "Pain & sadness. All is not well."

842 DATA "Examination. The ability to study details normally missed.", "Skill and bravery. The time to use such virtues."

850 DATA "Sadness or embarrassment for you (if female) or a woman close to you.", "The effects of one with authority to make decisions."

860 DATA "Contentment with all relationships.", "Love and passion. The inter-relation of the sexes."

862 DATA "A happy conclusion. Merriment and fulfilment.", "Tiredness and imaginary worries caused by a strict routine."

870 DATA "Receipt of a gift. This results in another loss.", "Reflections on the past and childhood."

872 DATA "Imagination. A good time for creative work though transitory.", "Dejection over things as they are. This will pass."

880 DATA "A victory in something you desire", "Contentment with your present surroundings and things as they are."

882 DATA "The effects of a fair & studious young man. He may help you.", "A proposition or invit



ation."

890 DATA "The effects of a fair woman. She is dreamy but visionary.", "The effects of a creative man in business, law or the church."

900 DATA "The beginning of new things - creation. Possibly a birth.", "Physical suffering or sadness. Dissatisfaction with possessions"

902 DATA "Strength in trade and dealings with people.", "A happy life. Peace & prosperity"

910 DATA "Imitation or sham. A superficial view.", "Success in undertakings or great news."

912 DATA "A need to work by discussion and negotiation. Final success.", "Haste towards a favourable outcome. Change."

920 DATA "Delay caused by opposition to your ideas.", "Financial gain but opposition in law."

922 DATA "The influence of a dark young man. He may bring news.", "Departure or absence. A move or emigration."

930 DATA "The influence of a country woman who is loving and honourable.", "The influence of a friendly countryman who is honest and conscientious."

940 DATA "Disquiet and a feeling that all is not right.", "The physical side of love. Passion. A very strong card."

942 DATA "Light and truth in matters. Complex problems revealed.", "Benevolence of others towards you."

950 DATA "An increase in you

understanding of society.", "Certain projects may be foolishly thought out."

952 DATA "A dispute. Possible litigation.", "A tendency to bigamy. Watch against bigotry."

960 DATA "An overcautious attitude. Think things out - then act.", "Abundance of the good things in life. Enjoy them."

962 DATA "An abuse of power through weakness.", "Obstruction through selfishness. Take heed."

970 DATA "Lethargy. Inaction could cause problems.", "Connections with the church or religion look well."

972 DATA "Blindness to what is right. Pettiness.", "Problems caused by argument or disagreement."

980 DATA "A tendency to arrogance which may cause alienation.", "Instability and inconstancy of those about you."

982 DATA "Contentment on material and spiritual planes."

990 DATA "Weakness through an oversimplification of events.", "Inertia or fixity may cause stagnation.", "Folly through extravagance or pleasure-seeking."

1000 DATA "Bad intelligence. The evil side of wealth.", "A facade of enjoyment to keep others happy."

1002 DATA "mediocrity in work. A tendency to pettiness.", "Delay to some project due to others' opposition."

1010 DATA "Disorder or chaos. It may spell ruin for some project.", "Envy and jealousy which lead to illusory desires."

1012 DATA "Quarrels over money or possessions.", "A tendency to vanity. This may stand against ambition."

1020 DATA "Deception. Beware of others keeping bad faith.", "Monetary loss through gambling or robbery. Take no chances."

1022 DATA "Unfavourable news causing worry.", "Idleness and lack of energy. Stagnation."

1030 DATA "Suspicion & mistrust may effect your reasoning.", "The temptation of vice. Watch for weakness."

1040 DATA "Great force in love or hate. Not to the good.", "Lies and disloyalty. Beware untruths."

1042 DATA "Confusion and a feeling of intellectual isolation.", "Precaution. Be careful what you do."

1050 DATA "Loss or destruction of something close to you.", "A proposal of love or other declaration. Publicity."

1052 DATA "Good advice from a wise person. Take heed.", "Difficulties and opposition. Treachery."

1060 DATA "Suspicion cast on you or a feeling of shame.", "Transitory profit or advantage. Make hay..."

1062 DATA "The use of observation to bad ends. Spying.", "Foolishness. The inability to thin

k soundly."

1070 DATA "A closed mind. Watch against bigotry or deceit.", "A tendency to cruelty or perversity. Watch this."

1080 DATA "Discontent. The desire to put right your problems.", "Love and passion. The inter-relation of the sexes."

1082 DATA "A new venture or holiday. Achievement from this.", "New relationships and a delight in novelty."

1090 DATA "The chance of new alliances. Business looks well.", "The chance of fresh excitement. Renewal."

1092 DATA "A desire to succeed. Your force of will.", "Great joy and happiness. A celebration."

1100 DATA "Loyalty of others to you. Honesty and good advice.", "Indignation at an affront in love. A violent response."

1102 DATA "A taste for risk could lead to deception.", "A trick or fraud. Be on your guard."

1110 DATA "The influence of a distinguished woman. Don't trust her.", "The effect of a rogue or scandalmonger. He is dishonest."

1120 DATA "A fall in fortune. Ruin of something established.", "Surprise and wonder leading to disquiet."

1122 DATA "The end of troubles. Adversity is over.", "Prosperity and increase. An appreciation of beauty."

```

1130 DATA "Contradiction and the
tendency of others to trick yo
u.", "Apprehension and the fea
r of defeat."
1132 DATA "Perplexity. Embarrass
ments and worries.", "Quarrels a
rising from jealousy or envy."
1140 DATA "Obstacles to your
ideas. Possible calamity.",
"Difficulties and the intrigue o
f others against you."
1142 DATA "Bad news or announcem
ents. Indecision.", "Discord
. Interruption of the present
status quo."
1150 DATA "Problems of jealous
y - deceit or infidelity.", "The
influence of a good but sev
ere man. He is austere, yet
tolerant."
1500 INVERSE 1: PRINT AT 21,0;"
Please press SPACE to continue "
: INVERSE 0
1510 IF INKEY$(">") THEN GO TO
1510
1520 RETURN
2000 RESTORE 2100
2010 FOR n=1 TO cn: READ y,x: PR
INT AT y,x;"A"
2020 IF cn=2 THEN PRINT AT y,x;
"B"
2030 NEXT n
2050 RETURN
2100 DATA 18,10,18,10,17,10,19,1
0,18,9,18,11,18,12,18,13,17,13,1
6,13
3000 RESTORE 3020: FOR n=1 TO 8:
READ row
3010 POKE USR "A"+n,row: NEXT n
3020 DATA 60,60,60,60,60,60,60,0
3030 FOR n=1 TO 8: READ row
3040 POKE USR "B"+n,row: NEXT n

```

```

1070 DATA 0,0,255,255,255,255,0,
1079 RETURN

```



# BALLOON

LEON GOODFRIEND  
Original program by Adrian Roe

## GENERAL DESCRIPTION

In this exciting game you must guide your hot air balloon through the maze which appears on the screen, avoiding the walls which will puncture it and cause you to crash. You must move your balloon from the bottom-right to the top-left of the screen. As you move, the wind can cause your basket to swing away from the balloon, which makes it even harder to avoid the walls.

## DETAILED DESCRIPTION

Lines 10 Set up screen colour and create space for the machine code subroutine.  
20 Call game initialisation subroutine.  
30 Set score, number of lives, and bonus.  
40 Call screen printing routine.  
50 Call balloon printing routine.  
60 Print score.  
70 Is right or left being pressed?  
80 Is up or down being pressed?  
100 Increase swing counter, bonus counter, and if swing counter = di then change swing of basket.  
105 If bonus counter > 30, decrease bonus.  
110 Change balloon's position.  
120 Is height a new maximum height?  
130 Erase balloon.  
140 Update balloon's x and y co-ordinates and repeat loop from line 50.  
1000-1370 Print the balloon and basket in the 4 different basket positions.

2000 If balloon has reached top-left then set up a new screen.  
2010-2080 Flash the screen four times.  
2090-2100 Reset screen to cyan.  
2110 Erase balloon.  
2120-2170 Print the basket as it falls to the ground.  
2180-2275 Print number of lives remaining.  
2280 Have you lost?  
2290-2300 You still have lives to spare so continue.  
3000-3010 Screen completed so beep and increase score.  
3020 If this is the first screen then restart game loop.  
3030 If you have already got the maximum number of lives then restart the game loop.  
3040-3080 Increase number of lives, bonus and print them on the screen.  
6000-6145 Print 'game over'.  
6150-6180 If another game is required then restart.  
8000-8540 Balloon has crashed or this is the 1st screen; so print screen, set swing, bonus, zero counters, print score and bonus.  
9000-9360 Print big balloon, give instructions if required.  
9800 POKE machine code into memory.  
9900-9980 Data for screen picture, machine code and user-defined graphics.

```
10 INK 0: PAPER 0: FLASH 0: BRIGHT 0:
OVER 0: INVERSE 0: BORDER 0: CLEAR 32512
20 GO SUB 9000
30 LET sc=0: LET bn=10000: LET lf=3: L
ET di=55: LET ll=0
40 GO SUB 8000
50 GO SUB sw
60 PRINT #0; AT 1,7; sc;
70 LET d=(IN 32766=251)-(IN 32766=247)
80 LET e=(IN 65022=254)-(IN 65278=253)
90 BEEP .1,0
100 LET t=t+1: LET u=u+1: IF t>di THEN
LET t=0: LET sw1=sw+100: IF sw1=1400 TH
EN LET sw1=1000
105 IF u>30 THEN LET u=0: LET bs=bs-25
```

```

%SGN bs: PRINT AT 0,0; OVER 0;"Bonus: ";
bs: CHR$ 143
110 LET p=x+5*d: LET q=y+5*e
120 IF q>ym THEN LET ym=q: LET sc=sc+2
0*11
130 GO SUB sw+70: LET sw=sw1
140 LET x=p: LET y=q: GO TO 50
1000 PLOT x,y: DRAW 1,0: DRAW 3,3: DRAW
0,3: DRAW -3,3: DRAW -3,0: DRAW -3,-3: D
RAW 0,-3: DRAW 3,-3: DRAW 1,0: DRAW -9,-
5: DRAW 2,-2: DRAW 2,2: DRAW -2,2
1010 IF POINT (x+1,y+9)+POINT (x-5,y+3)+
POINT (x+4,y+3)+POINT (x-8,y-7)+POINT (x
+4,y+6)+POINT (x-5,y+6)+POINT (x-2,y+9)+
POINT (x-10,y-5)+POINT (x-6,y-5)<>9 THEN
GO TO 2000
1020 RETURN
1070 PLOT x,y: DRAW 1,0: DRAW 3,3: DRAW
0,3: DRAW -3,3: DRAW -3,0: DRAW -3,-3: D
RAW 0,-3: DRAW 3,-3: DRAW 1,0: DRAW -9,-
5: DRAW 2,-2: DRAW 2,2: DRAW -2,2: RETUR
N
1100 PLOT x+1,y: DRAW 3,3: DRAW 0,3: DRA
W -3,3: DRAW -3,0: DRAW -3,-3:: DRAW 0,-
3: DRAW 3,-3: DRAW 2,0: DRAW 0,-6: DRAW
-2,-2: DRAW 2,-2: DRAW 2,2: DRAW -1,1
1110 IF POINT (x+1,y+9)+POINT (x-5,y+3)+
POINT (x+4,y+3)+POINT (x,y-10)+POINT (x+
4,y+6)+POINT (x-5,y+6)+POINT (x-2,y+9)+P
OINT (x+2,y-8)+POINT (x-2,y-8)<>9 THEN
GO TO 2000
1120 RETURN
1170 PLOT x+1,y: DRAW 3,3: DRAW 0,3: DRA
W -3,3: DRAW -3,0: DRAW -3,-3:: DRAW 0,-
3: DRAW 3,-3: DRAW 2,0: DRAW 0,-6: DRAW
-2,-2: DRAW 2,-2: DRAW 2,2: DRAW -1,1: R
ETURN
1200 PLOT x-1,y: DRAW 2,0: DRAW 3,3: DRA
W 0,3: DRAW -3,3: DRAW -3,0: DRAW -3,-3:
DRAW 0,-3: DRAW 3,-3: DRAW 1,0: DRAW 9,
-5: DRAW -2,-2: DRAW -2,2: DRAW 2,2
1210 IF POINT (x+1,y+9)+POINT (x-5,y+3)+

```

```

POINT (x+4,y+3)+POINT (x+8,y-5)+POINT (x
+4,y+6)+POINT (x-5,y+6)+POINT (x-2,y+9)+
POINT (x+4,y-5)+POINT (x+6,y-7)<>9 THEN
GO TO 2000
1220 RETURN
1270 PLOT x-1,y: DRAW 2,0: DRAW 3,3: DRA
W 0,3: DRAW -3,3: DRAW -3,0: DRAW -3,-3:
DRAW 0,-3: DRAW 3,-3: DRAW 1,0: DRAW 9,
5: DRAW -2,-2: DRAW -2,2: DRAW 2,2: RETURN
1300 GO TO 1100
1370 GO TO 1170
2000 IF x<44 AND y>138 THEN GO TO 3000
2010 FOR a=0 TO 4
2020 POKE 32512,7
2030 LET b=USR 32513
2040 BEEP .2,-10
2050 POKE 32512,56
2060 LET b=USR 32513
2070 BEEP .2,-10
2080 NEXT a
2090 POKE 32512,40
2100 LET a=USR 32513
2110 GO SUB sw+70
2120 LET p=INT (x/8): LET q=21-INT (y/8)
2130 FOR a=q TO 21
2140 PRINT AT a,p;"b";
2150 BEEP .3,21-a
2160 PRINT AT a,p;"b";
2170 NEXT a
2180 INVERSE 1: BRIGHT 1: OVER 0: CLS
2190 PRINT AT 10,7;"Lives remaining: ";l
f;AT 13,14;"aaaaaa" ( TO 1f)
2200 FOR a=1 TO 3: BEEP .1,15: PAUSE 40:
NEXT a
2210 LET lf=lf-1
2220 PRINT AT 10,24;lf
2225 INVERSE 0
2230 FOR a=7 TO 0 STEP -1
2240 PRINT INK a;AT 13,14+lf;"c"
2260 BEEP .1,0
2270 NEXT a
2272 PRINT AT 13,14+lf;" "

```

```

2275 PAUSE 100
2280 IF lf=0 THEN GO TO 6000
2290 LET di=di+10: LET li=li-1
2300 GO TO 40
3000 BEEP 1.2,2: BEEP 1.2,6: BEEP 1.4,9:
  BEEP .3,13: BEEP 1,12
3010 LET sc=sc+bs
3020 IF sc<bn THEN GO TO 40
3030 IF lf=6 THEN GO TO 40
3040 LET lf=lf+1
3050 LET bn=10000*(1+INT (sc/10000))
3060 PRINT INK 7; PAPER 0; OVER 0; AT 10
,7;" "; AT 11,7;" "; FLASH
1;"BONUS LIFE"; FLASH 0;" "; AT 12,7;"
"
3070 FOR a=1 TO 9: BEEP .1,12: NEXT a
3075 PAUSE 140
3080 GO TO 40
6000 INVERSE 1: CLS : INVERSE 0
6060 PLOT 28,110: DRAW 4,0: DRAW 0,-10:
DRAW 0,22,-1.5*PI
6070 PLOT 36,96: DRAW 12,30: DRAW 12,-30
: PLOT 40,106: DRAW 16,0
6080 PLOT 64,96: DRAW 0,30: DRAW 12,-12:
DRAW 12,12: DRAW 0,-30
6090 PLOT 116,96: DRAW -24,0: DRAW 0,30:
DRAW 24,0: PLOT 92,111: DRAW 20,0
6100 CIRCLE 148,111,15
6110 PLOT 165,126: DRAW 12,-30: DRAW 12,
30
6120 PLOT 217,96: DRAW -24,0: DRAW 0,30:
DRAW 24,0: PLOT 195,111: DRAW 20,0
6130 PLOT 221,96: DRAW 0,30: DRAW 14,0:
DRAW 0,-16,-PI: DRAW -14,0: DRAW 8,0: DR
AW 12,-14
6140 PRINT AT 15,7;"Final score: ";sc
6145 FOR a=20 TO 1 STEP -1: BEEP .05,a:
NEXT a
6150 PRINT "" "Another game -y/n "; FLAS
H 1;"?"
6160 IF INKEY$="n" OR INKEY$="N" THEN S
TOP

```

```

6170 IF INKEY$="y" OR INKEY$="Y" THEN C
LEAR : GO TO 30
6180 GO TO 6160
0000 RESTORE : INK 0: PAPER 5: INVERSE 0
1 BRIGHT 0: CLS
0010 READ a
0020 FOR b=1 TO a: PRINT CHR$ 143;: NEXT b
0030 READ a: IF a=0 THEN GO TO 8500
0040 FOR b=1 TO a: PRINT " ";: NEXT b
0050 GO TO 8010
0500 LET sw=1100: LET sw1=sw: LET bs=100
0: LET ym=0: LET u=0: LET t=0: LET di=di
-10: LET li=li+1
0510 LET x=231: LET y=27
0520 PRINT #0;"Score: ";sc,"aaaaaa"( TO
lf);
0525 PRINT AT 0,0;"Bonus: 1000"
0530 OVER 1
0540 RETURN
9000 GO SUB 9800
9005 PRINT INK 7;TAB 12;"THIS";: BEEP 1
.2,2: PAUSE 30: PRINT AT 7,12;"BALLOON!"
AT 0,17; INK 7;"IS": BEEP 1.2,6
9010 CIRCLE 127,117,50
9015 BEEP 1.4,9
9020 PLOT 81,97: DRAW 30,-50
9030 PLOT 174,97: DRAW -30,-50
9040 FOR a=111 TO 144 STEP 3: PLOT a,23:
DRAW 0,25: NEXT a
9050 FOR a=23 TO 48 STEP 5: PLOT 111,a:
DRAW 33,0: NEXT a
9060 BEEP .3,13
9070 POKE 32512,58: LET a=USR 32513
9080 BEEP 1,12
9090 PAPER 5
9100 FOR a=0 TO 23: READ b: POKE USR "a"
a,b: NEXT a
9110 PAUSE 100
9120 INPUT "Do you want instructions - y
/n"; LINE a$
9130 IF a$="n" OR a$="N" THEN RETURN
9140 IF a$<>"y" AND a$<>"Y" THEN GO TO

```



```

9120
9150 CLS
9160 PRINT "Your job is to fly your bal
loon"
9170 PRINT "from its start position at
the"
9180 PRINT "bottom right of the screen
to"
9190 PRINT "the top left. You must keep
to"
9200 PRINT "the blue sky because the bl
ack"
9210 PRINT "clouds hide many birds whic
h"
9220 PRINT "will burst your balloon."
9230 PRINT "Bonus points are awarded fo
r"
9240 PRINT "completing the course quick
ly."
9250 PRINT "'TAB 9;"Press any key";
9260 PAUSE 0
9270 CLS
9280 PRINT "'Your balloon is controlled
with"
9290 PRINT "the following keys:-"
9300 PRINT "'TAB 15;"A";'TAB 13;"N + M"
;'TAB 15;"Z"
9310 PRINT "'You can move diagonally by
"
9320 PRINT "pressing the appropriate tw
o"
9330 PRINT "movement keys together."
9340 PRINT "' Press any key to play"
;
9350 PAUSE 0
9360 RETURN
9600 RESTORE 9950: FOR a=32513 TO 32529:
READ b: POKE a,b: NEXT a: RETURN
9900 REM Sc
9910 DATA 65,10,1,3,18,15,5,7,6,26,6,27,
7,7,1,11,1,4,8,4,6,3,7,4,8,5,6,3,7,4,9,3
,6,3,4,8,8,2,2,19,8,23,8,20,1,2,8,11,21,

```

```

4,28,3,29,8,4,3,2,3,2,3,7,27,6,27,5,27,7
,15,2,6,35,0
9950 REM Machine code data
9960 DATA 33,0,88,58,0,127,119,17,0,86,1
,1,0,3,237,176,201
9970 REM Character data
9980 DATA 56,68,130,130,66,56,56,0,0,0,6
0,66,36,36,24,0,36,36,195,24,24,195,36,3
0

```

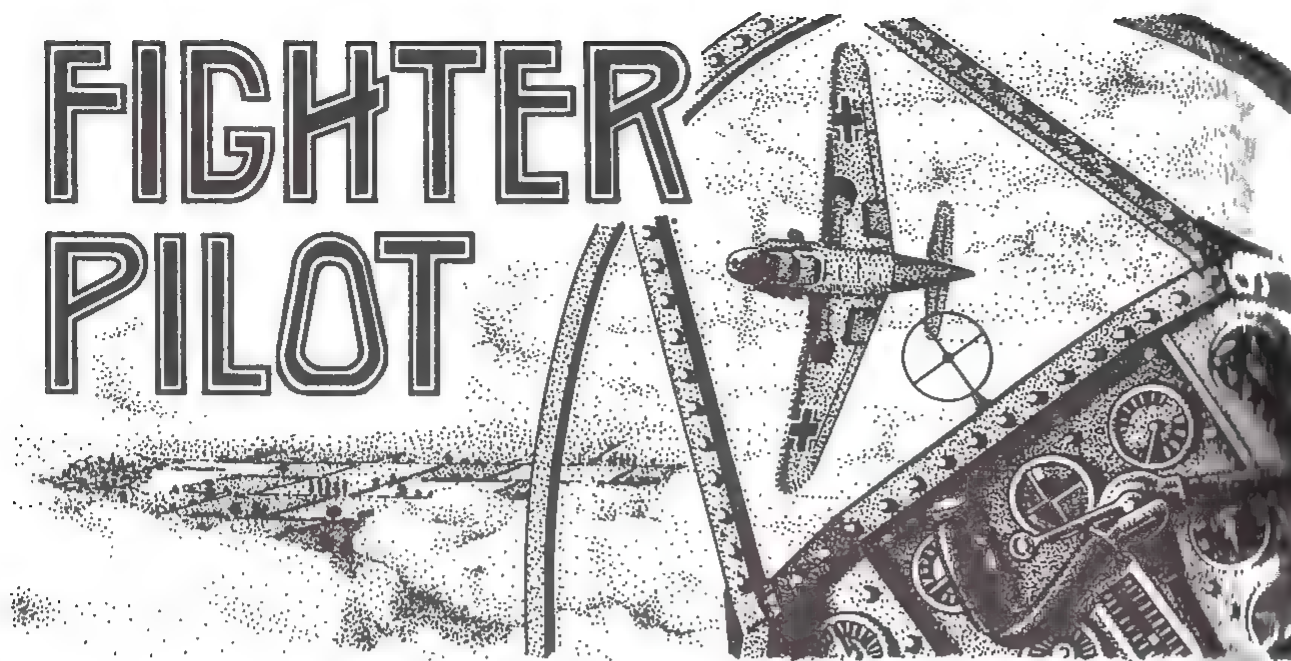
Some models of Spectrum  
will not RUN this  
program. If you experience  
difficulty, replace lines  
70 and 80 with:

```

70 LET d=(IN 32766=187)-(IN 32
766=183)
80 LET d=(IN 65022=190)-(IN 65
278=189)

```

# FIGHTER PILOT



DAVID JONES

Original program by E.G. Kemplen

## GENERAL DESCRIPTION

You must earn your wings in this contest of aerial combat. To enact the Battle of Britain as you struggle to line up the enemy M.E.109 plane in your Spitfire's gunsights, then stab at the fire button. Depending on your aim you may hit his wings, tail or fuselage, taking you nearer to your target of 50 hits. However there is no time to relax, for as soon as you shoot one plane down another appears in your field of view.

This game features nine difficulty levels and you must keep a check on your fuel and ammo supplies as they are not limitless.

## DETAILED DESCRIPTION

Lines 50-70 Assign variables and set up the user-defined graphics characters.

80-170 Set up the screen display.

180-200 Position the enemy plane.

220-260 Main program loop and user input.

270 Out of fuel.

280 Out of ammunition.

290 Game over?

300-390 Move enemy plane at random then jump back to main loop.

400-490 Act on user's input.

500-560 Hit routine.

570-590 Print a new enemy plane.

600-650 Game over. Do you want to play again?

670-860 Instructions.

870-910 Input difficulty level.

920-990 Data for graphics.

1010-1030 Shot subroutines.

```

10 REM Fighter pilot
20 REM By E G Kemplen
30 REM Converted to Spectrum
40 REM by David Jones 27/07/83
50 FOR i=0 TO 63: READ a: POKE USR "a"
+1,a: NEXT i
60 POKE 23658,8: REM set CAPS
70 LET s=0: LET sc=0: LET f1=0: LET ax
=0: LET ay=0
80 INK 0: PAPER 6: CLS : BORDER 6: INP
UT ""
90 GO SUB 670
100 INK 0: PAPER 5: CLS : BORDER 0: INP
UT ""
110 PLOT 123,16: DRAW 0,159
120 PLOT 124,16: DRAW 0,159
130 PRINT AT 20,0; PAPER 4;"
      Fighter                                     Pilot
                                     ";
140 PRINT AT 20,0; PAPER 4;"Score =" ;sc
|
150 PRINT AT 21,0; PAPER 4;"Ammo =" ;100
-5;
160 PRINT AT 21,22; PAPER 4;"Fuel =" ;10
00-f1;
170 LET cx=15: LET cy=12: PRINT AT cy,c
x;"9";
180 LET bx=15: LET by=3
190 REM Position enemy

```

```

200 PRINT AT by,bx; OVER 1; INK 1;"f";A
T by+1,bx-2;"abcde"
210 REM Keyscan
220 IF df>0 THEN PAUSE df
230 LET x$=INKEY$: IF x$>="A" AND x$<="
Z" THEN GO SUB 400: GO TO 300
240 IF x$=" " THEN LET b7=ATTR (cy,cx)
: GO SUB 1010: LET s=s+1: PRINT AT 21,6;
PAPER 4;" ";AT 21,0;"Ammo =" ;100-s;
250 IF x$=" " THEN IF b7=41 THEN GO S
UB 500
260 LET f1=f1+1: PRINT AT 21,28; PAPER
4;" ";AT 21,22;"Fuel =" ;1000-f1;
270 IF f1=1000 THEN PRINT AT 0,0; FLAS
H 1;"Out of fuel": GO TO 610
280 IF s=100 THEN PRINT AT 0,0; FLASH
1;"Out of ammunition.": PRINT "Run like
hell !": GO TO 610
290 IF sc>=50 THEN GO TO 600
300 LET nx=INT (RND*3)-1: LET ny=INT (R
ND*3)-1
310 LET dy=by: LET dx=bx
320 LET bx=bx+nx+ax: LET by=by+ny+ay
330 IF by>18 THEN LET by=18
340 IF by<0 THEN LET by=0
350 IF bx>29 THEN LET bx=29
360 IF bx<2 THEN LET bx=2
370 PRINT AT dy,dx; OVER 1; INK 1;"f";A
T dy+1,dx-2;"abcde"
380 PRINT AT dy,15; INK 0;"h";AT dy+1,1
5;"h";AT cy,cx;"g";
390 GO TO 200
400 IF x$="Q" THEN LET ax=1: LET ay=1:
RETURN
410 IF x$="W" THEN LET ax=0: LET ay=1:
RETURN
420 IF x$="E" THEN LET ax=-1: LET ay=1
: RETURN
430 IF x$="A" THEN LET ax=1: LET ay=0:
RETURN
440 IF x$="S" THEN LET ax=0: LET ay=0:
RETURN

```

```

450 IF x$="D" THEN LET ax=-1: LET ay=0
: RETURN
460 IF x$="Z" THEN LET ax=1: LET ay=-1
: RETURN
470 IF x$="X" THEN LET ax=0: LET ay=-1
: RETURN
480 IF x$="C" THEN LET ax=-1: LET ay=-
1: RETURN
490 LET ax=0: LET ay=0: RETURN
500 LET ii=5: IF by=cy THEN LET ii=2
510 IF by=cy-1 AND bx<>cx THEN LET ii=
2
520 LET sc=sc+ii: FOR b=62 TO 24 STEP -
6: BEEP .022,b: NEXT b
530 PRINT AT 20,0; PAPER 4;"Score =" ;sc
:
540 PRINT AT cy-1,cx-4;" h ";
550 PRINT AT cy,cx-4;" g ";
560 PRINT AT cy+1,cx-4;" h ";
570 LET bx=15: LET by=3
580 PRINT AT by,bx; OVER 1; INK 1;"f";A
T by+1,bx-2;"abcde"
590 RETURN
600 PRINT AT 0,0;"Enemy planes destroye
d using's;' rounds of ammunition."
610 PRINT "To play again press 'SPACE'."
"
620 PRINT "To end press 'E'."
630 PAUSE 0: LET a$=INKEY$: IF a$=" " T
HEN RUN 60
640 IF a$<>"E" THEN GO TO 630
650 STOP
660 REM Instructions
670 PRINT "
"
680 PRINT "You are the pilot of a SPITF
IRE."
690 PRINT "Ahead of you is an M.E.109."
700 PRINT "He is taking avoiding action
,"

```



```

710 PRINT "and you must try to shoot hi
m   down."
720 PRINT "Your guns are fired by press
ing"
730 PRINT "the 'SPACE' key."
740 PRINT "The letters 'QWEASDZXC' are
used";
750 PRINT "to control your SPITFIRE."
760 PRINT "The top row 'QWE' make your"
770 PRINT "plane climb and the bottom r
ow 'ZXC' make your plane dive."
780 PRINT "The left column 'QAZ' make y
our plane turn left and the right"
790 PRINT "column 'EDC' will turn you t
o   the right."
800 PRINT "PRESS ANY KEY TO  CONTINUE "
: PAUSE 0: LET z$=INKEY$
810 CLS : PRINT "

```

```

"
820 PRINT "The 'S' key will straighten
the plane up."
830 PRINT "You cannot alter course whil
st firing your guns."
840 PRINT "The object of the game is to
"
850 PRINT "score 50 points."
860 PRINT "A hit on the wing or tail sc
ores 2 and a fuselage hit scores 5."
870 PRINT "'Press 0 to 9 for difficult
y'" "level and the game will begin."
880 PRINT "'0 = Easy 9= Hard."
890 PAUSE 0: LET a$=INKEY$: IF a$<"0" O
R a$>"9" THEN GO TO 890
900 LET df=(9-VAL (a$))*10
910 RETURN
920 DATA 0,0,0,3,63,0,0,0
930 DATA 0,0,0,13,255,0,0,0
940 DATA 28,62,255,255,255,255,62,28
950 DATA 0,0,128,216,255,128,0,0
960 DATA 0,0,0,96,254,0,0,0

```

```

970 DATA 0,0,8,8,8,8,8,28
980 DATA 0,24,24,231,231,24,24,0
990 DATA 24,24,24,24,24,24,24,24
1000 REM Shot subroutine
1010 FOR i=1 TO 3: FOR j=0 TO 7: LET k=j
+3: IF k>7 THEN LET k=k-7
1020 PRINT AT cy,cx; OVER 1; INK j; PAPE
R k;" "; BEEP .01,10: NEXT j: NEXT i
1030 PRINT AT cy,cx; INK 0; PAPER 5; OVE
R 0;"g";: RETURN

```

# BOOK INDEX

ANDREW CRESSWELL  
Original program by Ian Andrews

## GENERAL DESCRIPTION

This program will produce an alphabetical index of the data entered. First of all you must enter the name of the book, and its author, then you make a decision on the number and length of entries for the index data. Make allowances for possible future additions as you cannot easily alter these figures.

Next, you are asked to type in the "text" part of the entry, followed by its page number, until you have entered all of the data. The computer will sort this information into alphabetical order. You then have the option of printing the index on a printer, saving it on cassette, starting a new index list, or ending the program.

N.B. The 48K Spectrum will have a greater capacity for data storage than the 16K version but in both cases the computer checks that it has sufficient memory to store your information. The greater the amount of data entered, the longer the sorting process will last.

## DETAILED DESCRIPTION

Lines 1-40 Titles.

50 Function for use when centering strings on display.

70-120 Entry of name of book.

150-230 Establish size of index and ensure sufficient memory is available to fit.

240-265 Prepare for entry of index.

270-480 Input entries of appropriate length, and page number for reference.

500-580 Sort references into alphabetical order.

610-700 Print index.

710-735 Offer options for printing, saving, or erasing index,

or even exiting the program.

740-770 Goto appropriate routine for selected option.

780 Save option.

810-910 Printer option, print out index on printer.

```
1 REM *****
2 REM **** Book Index ****
3 REM *****
4 REM Original ZX81 program
5 REM by Ian Andrews
6 REM *****
7 REM Spectrum conversion
8 REM by Andrew Cresswell
9 REM *****
40 CLEAR
50 DEF FN t(x$)=INT ((32-LEN x$)/2)
70 PRINT TAB 10; INK 7; PAPER 2;"I N D
E X";AT 2,0; PAPER 1;"Enter title": INP
UT a$
120 PRINT AT 2,0; INK 7; PAPER 1;"Enter
author": INPUT b$
150 PRINT AT 2,0; INK 7; PAPER 1;"Enter
an estimate of the number of entries yo
u will make."
160 PRINT AT 15,0; INK 1; PAPER 7;"NB-B
e generous as you cannot extend the n
umber later on."
170 INPUT n
190 PRINT AT 2,0; INK 7; PAPER 1;"Enter
max length of entries."; PAPER 7;"
": INPUT m
210 LET pramt=PEEK 23732+256*PEEK 23733
: LET ext=pramt-32767
220 LET d=5000+ext-3*n
230 IF d<m*n THEN PRINT AT 2,0; INK 7;
PAPER 2;" Re-enter no. of entries -
Max with ";m;" characters is ";INT (d
/m): GO TO 170
240 DIM l$(n+1,m+3)
260 CLS : PRINT INK 1;"Get ready to en
ter words one by one (up to "; INK 2;"**
";m;"**"; INK 1;" characters","in length
```

```

). "
265 PRINT "'Press any key when ready"
270 PAUSE 0: PRINT : CLS
300 FOR i=1 TO n
310 PRINT INK 7; PAPER 1;"Entry ";i;AT
15,7;"Text?"
320 IF i=n-10 THEN PRINT AT 15,0; INK
7; PAPER 2;"WARNING - memory short."
330 INPUT x$: IF LEN x$<=m THEN GO TO
390
340 PRINT AT 8,5; INK 7; PAPER 2;"Abbre
viate entry"
350 PRINT AT 4,5; INK 1;x$;"
      ": FOR f=1 TO m: PR
INT AT 5,4+f;"-": NEXT f: GO TO 330
390 LET l$(i)( TO m)=x$: LET x$=" ": IF
CODE l$(i)=226 THEN LET n=i-1: GO TO 5
00
430 CLS : PRINT INK 1;"Entry ";i;TAB 1
2;l$(i)( TO m);AT 15,0; INK 7; PAPER 1;"
Page number?"
440 INPUT p$: LET c=CODE p$: LET s=1
450 IF c=32 THEN LET s=s+1: LET c=CODE
p$(s): GO TO 450
460 IF c<48 OR c>57 THEN PRINT AT 18,5
; INK 7; PAPER 2;"ERROR": GO TO 440
470 LET l$(i)(m+1 TO )=p$
480 CLS : NEXT i
500 REM          order
510 CLS : PRINT AT 10,12; PAPER 6; INK
2; FLASH 1;"Sorting"
520 FOR k=1 TO n-1: FOR j=1 TO n-k: IF
l$(j)<=l$(j+1) THEN GO TO 580
550 LET t$=l$(j): LET l$(j)=l$(j+1): LE
T l$(j+1)=t$
580 NEXT j: NEXT k
610 REM          print
630 CLS : PRINT TAB FN t(a$+" "); INK
1;"";a$;"""TAB FN t("by");"by"TAB F
N t(b$);b$: PRINT
640 PAUSE 100: PRINT
650 FOR i=1 TO n: PRINT INK 1;l$(i)( T

```

```

0 m);TAB m+7;l$(i)(m+1 TO ): NEXT i
700 PRINT "'TAB FN t("Index ends"); INK
7; PAPER 1;"Index ends"
710 INK 7: PAPER 1: PRINT "'Do you wan
t..."
712 PRINT "      S - to save it
      "
714 PRINT "      P - to print it
      "
716 PRINT "      A - to start a new index
      "
718 PRINT "or E - to end?
      "
720 POKE 23658,8: INK 0: PAPER 7
730 INPUT q$: IF q$<>"S" AND q$<>"P" AN
D q$<>"A" AND q$<>"E" THEN GO TO 730
735 POKE 23658,0
740 IF q$="S" THEN GO TO 780
750 IF q$="P" THEN GO TO 810
760 IF q$="A" THEN GO TO 30
770 GO TO 9999
780 SAVE "book index" LINE 600: CLS : G
O TO 710
810 CLS
820 LPRINT TAB FN t(a$+" ");"";a$;"
""TAB FN t("by");"by"TAB FN t(b$);b$:
LPRINT : LPRINT
850 LPRINT TAB 10;"I N D E X": LPRINT :
LPRINT
880 FOR i=1 TO n: LPRINT l$(i)( TO m);T
AB m+7;l$(i)(m+1 TO ): NEXT i
910 LPRINT : LPRINT : LPRINT TAB FN t("
Index ends");"Index ends": GO TO 710

```





ANDREW ESMOND  
Original program by P. Baker

## GENERAL DESCRIPTION

In this game you are trapped in a graveyard at midnight, and the Zombies are crawling out of their graves to try and catch you. Your task is to run around the graveyard trying to make all the Zombies fall back into their graves. If you fail and you get caught by a Zombie you will be 'Zombified', which is a fate worse than death ...

This game demonstrates vividly the colour, sound and user definable graphics capabilities of the Spectrum.

## DETAILED DESCRIPTION

Lines 10-20 Read in data about the directions of motion.

100-220 Instructions routine.

1000-1120 Main program loop routine which controls access to all the subroutines making up the program.

1500-1550 Get the player to specify the number of Zombies.

2000-2090 Draw the graveyard on the screen.

3000-3080 Get key being pressed from keyboard and move your marker on the screen accordingly.

3085-3150 Move each of the Zombies in turn.

3500-3570 Check to see if Zombie falls into a grave.

3700-3710 You have been caught!

4000-4050 See how long you have survived and check if it is a new highscore.

9000-9900 Set up user-defined graphics and highscore.

```

10 DIM A(10): DIM B(10): FOR I
11 TO 9: READ A(I),B(I): NEXT I
20 DATA -1,1,0,1,1,1,-1,0,0,0,
1,0,-1,-1,0,-1,1,-1
100 PAPER 0: INK 6: BORDER 0: C
10
105 GO SUB 9000
110 PRINT "                ZOMBIES";
AT 0,11: OVER 1: "_____ "
120 PRINT : PRINT "      YOU ARE I
N A GRAVEYARD,THE      CHURCH CLOCK
HAS JUST STRUCK      MIDNIGHT!!!"
130 PRINT "YOU MUST EVADE THE Z
OMBIES FOR      AS LONG AS POSSIBLE
BY LURING      THEM INTO OPEN GRAVE
S (AB).";
140 PRINT "BUT,IF YOU ARE CAUGH
T THEN YOU ARE      ";TAB 10; FLASH
1;"ZOMBIFIED"; FLASH 0
150 PRINT : PRINT "KEYBOARD COM
MANDS:"
160 PRINT "      Q W E"
170 PRINT "      \:/ "
180 PRINT "      A-S-D"
190 PRINT "      /\\"
200 PRINT "      Z X C"
205 PRINT "CAPS LOCK MUST BE ON
.PLEASE PUT IT ON AND PRESS 'ENT
ER'": INPUT A$
210 PRINT : PRINT "PRESS 'SPACE
' TO PLAY "; FLASH 1;"ZOMBIES!"
220 IF INKEY#(">") " THEN GO TO
220
1000 BRIGHT 1: CLS
1010 LET SW=32: REM SCREEN WIDTH

```

```

1020 LET SL=22: REM SCREEN LENGT
H
1030 LET SS=22528: REM ATTRIBUTE
START
1040 DIM Z(50,2)
1050 GO SUB 1500: REM INITIALIZE
1060 GO SUB 2000: REM SET UP BOA
RD
1070 GO SUB 3000: REM PLAY GAME
1080 INPUT AT 0,0;"ANOTHER GAME?
";A$
1090 IF A$="" THEN INPUT A$: GO
TO 1090
1100 IF A$(1)="Y" THEN GO TO 10
50
1110 IF A$(1)<>"N" THEN GO TO 1
080
1120 INK 7: CLS : STOP
1500 INPUT AT 11,0;"HOW MANY ZOM
BIES (20-50)";Z
1510 IF Z>50 THEN PRINT "TOO MA
NY": GO TO 1500
1520 IF Z<20 THEN PRINT "TOO FE
W": GO TO 1500
1540 LET ZZ=Z
1550 RETURN
1999 REM DRAW BOARD...
2000 CLS : PRINT "CDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDE"
2010 LET A$="F
I"
2020 LET B$="F AB AB AB AB
AB AB AB I"
2030 PRINT A$: FOR I=1 TO 6: PRI
NT A$: PRINT B$: PRINT A$: NEXT
I: PRINT A$
2040 PRINT "GDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDH"
2045 LET O0=15: LET O1=11: LET O
=SS+O0+SW*O1: LET D=0: LET J=1:
LET Z=ZZ: LET NS=1: LET NT=1
2050 PRINT AT O1,O0;"J": POKE O,

```

```

3: FOR I=1 TO Z
2060 LET P1=INT (((SW-3)*RND))+1
1 LET P2=INT ((SL-2)*RND)+1: LET
P=P1+P2*32+SS
2070 IF SCREEN$(P2,P1)<>" " THE
N GO TO 2060
2080 LET Z(I,1)=P1: LET Z(I,2)=P
2: PRINT AT P2,P1;"#": POKE P,19
2090 NEXT I: RETURN
2799 REM GAME...
3000 POKE 23672,0: POKE 23673,0:
POKE 23674,0
3010 LET A$=INKEY$
3011 LET V=10
3020 IF A$="" THEN GO TO 3070
3030 IF A$="Q" THEN LET V=7
3031 IF A$="W" THEN LET V=8
3032 IF A$="E" THEN LET V=9
3033 IF A$="A" THEN LET V=4
3034 IF A$="S" THEN LET V=5
3035 IF A$="D" THEN LET V=6
3037 IF A$="Z" THEN LET V=1
3038 IF A$="X" THEN LET V=2
3039 IF A$="C" THEN LET V=3
3040 LET D0=A(V): LET D1=B(V): L
ET D=D0+D1*SW
3070 IF SCREEN$(O1+D1,O0+D0)="
" THEN PRINT AT O1,O0;" ";AT O1
+D1,O0+D0;"J": LET O=O+D: LET O1
=O1+D1: LET O0=O0+D0: POKE O,3
3080 IF PEEK (O+D)=195 THEN PRI
NT AT O1,O0;" ": LET O=O+D: GO T
O 3700
3085 FOR I=1 TO NT
3090 LET J=INT (RND*Z)+1
3095 LET P=SS+Z(J,1)+Z(J,2)*SW:
POKE P,7: PRINT AT Z(J,2),Z(J,1)
1" "
3100 LET Z(J,1)=Z(J,1)-SGN (Z(J,
1)-O0)
3110 LET Z(J,2)=Z(J,2)-SGN (Z(J,

```

```

2)-01)
3120 LET P=SS+Z(J,1)+(Z(J,2)*SW)
3130 IF PEEK P=3 THEN GO TO 3700
3140 IF CODE SCREEN# (Z(J,2),Z(J,1))=0 THEN GO TO 3500
3150 PRINT AT Z(J,2),Z(J,1);"#": POKE P,195: NEXT I: GO TO 3010
3500 FOR F=1 TO 0 STEP -.1: BEEP .01,F: NEXT F
3510 IF CODE SCREEN# (Z(J,2),Z(J,1))=0 THEN LET P=P-1: LET Z(J,1)=Z(J,1)-1: GO TO 3510
3520: LET Z(J,1)=Z(Z,1)
3530 LET Z(J,2)=Z(Z,2)
3550 LET Z=Z-1
3560 IF Z<1 THEN GO SUB 2000: LET NS=NS+1: LET NT=NS: LET Z=Z+INT (RND*5): IF Z>49 THEN LET Z=49
3570 GO TO 3010
3700 FOR F=1 TO 15: BEEP .1,0: PRINT AT 01,00; FLASH 1; "#": BEEP .2,-2: PRINT AT 01,00; "J": NEXT F
3710 IF INKEY#="" THEN GO TO 3730
4000 LET T=FN T(): PRINT AT 1,0; "YOU HAVE SURVIVED FOR ";T;" SECS."
4010 IF T<HS THEN GO TO 4040
4020 PRINT "YOU HAVE SURVIVED LONGEST.": IF HS=0 THEN LET HS=T: GO TO 4050
4030 PRINT "THE PREVIOUS BEST TIME WAS ";HS: LET HS=T: GO TO 4050
4040 PRINT "THE BEST TIME IS STILL ";HS;" SECS."
4050 RETURN
9000 REM USER DEFINED GRAPHICS...
9010 RESTORE 9025: FOR F=0 TO 7:

```

```

9020 READ A,B: POKE USR "A"+F,A: POKE USR "B"+F,B: NEXT F
9025 DATA 12,0,63,192,127,252,12,255,127,255,127,252,63,192,12,
9030 FOR F=0 TO 7: READ C,D,E: POKE USR "C"+F,C: POKE USR "D"+F,D: POKE USR "E"+F,E
9040 NEXT F
9050 DATA 0,0,0,0,0,0,0,0,0,0,0,15,255,240,8,0,16,8,0,16,8,0,1
9060 FOR F=0 TO 7: READ G,H: POKE USR "F"+F,G: POKE USR "G"+F,H: POKE USR "H"+F,H: POKE USR "I"+F,16: NEXT F
9070 DATA 8,16,8,16,8,16,8,16,15,240,0,0,0,0,0,0
9080 FOR F=0 TO 7: READ J: POKE USR "J"+F,J: NEXT F
9090 DATA 0,60,126,126,126,126,60,0
9100 DEF FN T()=INT (65536*PEEK 23674+256*PEEK 23673+PEEK 23672)/50
9110 LET D0=0: LET D1=0: LET O0=0: LET O1=0
9120 LET HS=0
9900 RETURN

```



# AIRCRAFT LANDER

ANDREW ESMOND  
Original program by P. Bailey

## GENERAL DESCRIPTION

You are required to take control of a large aircraft with 300 people on board, and land it at the airport, seen through your cockpit window. You have various instruments to help you, displaying airspeed, bearing, pitch, roll, altitude, distance from runway and deviation from runway.

You can use the throttle to increase or decrease airspeed. Other controls allow you to roll left or right, dive or climb. Once you have landed on the 2000m long runway you must brake and use the rudder to try to remain on the runway.

Throughout the simulation you can see the runway lights piercing the gloom, and beyond, the lights of a distant city. You can alter the difficulty of landing by choosing wind speed and direction. Happy landings!

## DETAILED DESCRIPTION

Lines 90-340 Various utility subroutines.

100-110 Short delay.

200-210 Long delay.

300-340 User prompt for next page.

400-440 Update in-flight variables.

1000-1060 Main program loop approaching the airfield.

1200-1290 Main program loop after landing on the runway

1400-1500 Fatal crash routine.

1600-1690 Landed O.K. but left the runway.

1800-1820 Congratulatory message for safe landing.

2000-2140 Update in-flight position and speed.

3000-3040 Check for a crash.

4000-4270 Update the screen display.

5000-5110 User input - in-flight controls.

5500-5580 User input - after landing controls.

6000-6090 Update variables after landing.

7000-7060 Explosion on crashing.

8000-8640 Initialisation, set variables, display instructions and user input of wind strength and direction.

9000-9090 The black box, showing final values after crashing.

```
1 REM ***Aircraft landing simulator***
10 GO TO 8000
90 REM ***Utility subroutines***
100 REM ***Short delay***
110 FOR j=1 TO 50: NEXT j: RETURN
200 REM ***Long delay***
210 FOR j=1 TO 500: NEXT j: RETURN
300 REM ***End of page***
310 PRINT FLASH 1;"Press <ENTER> to continue"
320 LET x$=INKEY$: IF x$<>CHR$(13) THEN GO TO 320
330 CLS
340 RETURN
400 REM ***Calculate components of velocity***
410 LET ra=ad*ci
420 LET vx=as*SIN ra
430 LET vy=as*COS ra
440 RETURN
1000 REM ***Approach to airfield***
1010 GO SUB 5000
1020 GO SUB 2000
1030 IF pz<=0 THEN GO TO 1200
1040 GO SUB 4000
1050 GO SUB 5000
1060 GO TO 1010
```

```

1200 REM ***After touchdown***
1210 GO SUB 3000
1220 IF lf=1 THEN GO TO 1400
1240 GO SUB 5500
1250 GO SUB 6000
1260 GO SUB 4000
1270 IF lf=2 THEN GO TO 1600
1280 IF lf=3 THEN GO TO 1800
1290 GO TO 1250
1400 REM ***Fatal crash***
1410 PRINT AT 20,10;"Press any key"
1420 LET x$=INKEY$: IF x$<>" " THEN GO TO 1480
1430 PRINT AT 10,10; BRIGHT 1; FLASH 1;"FATAL CRASH";: GO SUB 9097
1440 GO SUB 110
1470 GO TO 1420
1480 CLS : PRINT "The flight recorder has been recovered. The final flight"
1490 PRINT "details were as follows :-"
1500 GO TO 9000
1600 REM ***Off runway after good landing***
1620 IF py>0 THEN PRINT AT 15,1;"You have gone off the end of"; AT 16,1;"the runway."; AT 17,1;"Try using your brakes next"; AT 18,1;"time!": GO SUB 210
1630 IF ABS px>wr THEN PRINT AT 15,1;"You are off the side of the "; AT 16,1;"runway. "; AT 17,1;"Isn't it wide enough for you? "; AT 18,1;" ": GO SUB 210
1640 IF as<=10 THEN PRINT TAB 1;"Luckily there is only minor"; TAB 1;"damage at this speed.": GO SUB 210
1650 IF as>10 AND as<=25 THEN P

```

```

PRINT AT 15,1;"Serious damage to your aircraft"; AT 16,1;"but you survive to scare your"; AT 17,1;"passengers to death again.": GO SUB 210
1660 IF as>25 THEN PRINT AT 19,1;"DISASTER-FATAL CRASH!!!": GO SUB 9097: GO SUB 210
1670 CLS
1680 PRINT "Details of the flight when you left the runway:-"
1690 GO TO 9000
1800 REM ***Good landing***
1810 PRINT AT 15,1;"Congratulations Captain."; AT 16,1;"A safe landing.": GO SUB 9095
1820 GO TO 9080
2000 REM ***Update position and speed in air***
2010 LET pz=pz+gz
2020 LET py=py+gy
2030 LET px=px+gx
2040 LET pt=pt+nc
2050 LET rl=rl+bc
2060 LET as=as+2*SGN (t(tc+1)-as)
2070 LET ad=ad+rl
2080 LET vz=1*(tc-5)+as*SIN (pt*ci)
2090 GO SUB 400
2100 LET gz=vz
2110 LET gy=vy+wy
2120 LET gx=vx+wy
2130 LET gd=-ATN (vx/vy)*c2
2140 RETURN
3000 REM ***Check for crash***
3010 IF ABS rl>rt OR pt>tp OR pt<0 OR ABS ad>yt OR as<45 OR as>55 THEN LET lf=1: PRINT AT 15,1;"Crash on landing";: GO SUB 7000
3020 IF ABS px>wr OR py>0 OR py<-1r THEN LET lf=1: PRINT AT 16,

```

```

1;" off runway!";
3030 GO SUB 200
3040 RETURN
4000 REM ***Update display***
4010 LET ht=-pz-5: LET ld=-px-wr
: LET rd=-px+wr
4020 LET hd=ht*7: IF hd<-py THEN
    LET hd=-py
4030 IF hd<3000 THEN LET hd=3000
4040 FOR i=1 TO 5
4050 LET yd=-py-i*500
4060 LET q(i,1)=ld: LET q(i+5,1)
    =rd
4070 LET q(i,2)=yd: LET q(i+5,2)
    =yd
4080 LET q(i,3)=ht: LET q(i+5,3)
    =ht
4090 NEXT i
4100 LET q(11,1)=-px: LET q(11,2)
    =-py: LET q(12,3)=ht
4110 LET q(12,1)=-px: LET q(12,2)
    =-py-1r: LET q(12,3)=ht
4120 LET a1=64/wr: LET a2=63.5:
    LET a3=15.5: LET a4=31
4130 FOR i=1 TO 12
4140 LET a0=(hd-q(i,2))/(hd*2)
4150 LET r(i,1)=INT (q(i,1)*a0*a
    1+a2)
4160 LET r(i,2)=INT (a3+a0*a4)
4170 NEXT i
4190 FOR i=7 TO 20
4192 PAPER 0: PRINT AT i,0;"
    "
4194 NEXT i
4200 FOR i=1 TO 12
4210 PLOT 40+r(i,1),ABS (90-r(i,
    2))
4220 NEXT i
4260 PRINT AT 1,0: PAPER 1:INT (
    as+c5);TAB 10;INT (ad+c5);TAB 20
    ;INT (pt+c5);TAB 27;INT (r1+c5);

```

```

" ";AT 3,0;INT (pz+c5);TAB 10;I
NT (ABS py+c5);TAB 20;INT (px+c5
);" "
4270 RETURN
5000 REM ***Check controls***
5010 LET x$=""
5020 LET x$=INKEY$
5030 IF x$="" THEN RETURN
5040 IF CODE x$>47 AND CODE x$<5
    0 THEN LET tc=VAL x$
5050 IF x$="F" THEN LET bc=-1
5060 IF x$="G" THEN LET bc=0
5070 IF x$="H" THEN LET bc=1
5080 IF x$="M" THEN LET nc=1
5090 IF x$="K" THEN LET nc=0
5100 IF x$="O" THEN LET nc=-1
5110 RETURN
5500 REM ***Controls after landi
    ng***
5510 LET x$=""
5520 LET x$=INKEY$
5530 IF x$="" THEN RETURN
5540 IF x$="R" THEN LET rc=1
5550 IF x$="E" THEN LET rc=-1
5560 IF x$="C" THEN LET rc=0
5570 IF x$="B" THEN LET fd=2
5580 RETURN
6000 REM ***After touchdown***
6010 LET pt=0: LET r1=0: LET pz=
    0
6020 GO SUB 5500
6030 LET px=px+vx
6040 LET py=py+vy
6050 IF ABS px>wr OR py>0 THEN
    LET lf=2
6060 LET ad=ad+ad*(SGN ad*SGN rc
    )
6070 LET as=as-fd: IF as<=0 THEN
    LET lf=3
6080 GO SUB 400
6090 RETURN
7000 REM *** explosion ***

```



```

7010 LET aa=0
7020 FOR i=255 TO 0 STEP -1
7030 OUT 254,aa: BEEP .01,RND*20
7040 LET aa=aa+15-256*(aa>238)
7050 NEXT i
7060 RETURN
8000 REM ***Initialisation***
8010 DIM q(12,3): DIM r(12,2)
8020 PAPER 1: INK 6: BORDER 1
8030 CLS : PRINT TAB 3;"Aircraft
  landing simulator": PRINT
8035 POKE 23658,8: REM set caps
8040 PRINT "Do you require instr
  uctions      (Y/N)? ": INPUT x$
8050 PRINT
8060 IF x$="N" THEN GO TO 8290
8065 IF x$="Y" THEN GO TO 8080
8070 INPUT "Please type Y or N";
x$: GO TO 8060
8080 CLS
8090 PRINT "It is dark and raini
  ng. You are approaching the airp
  ort with 300"
8095 PRINT "passengers on board
  and your co-pilot has just becom
  e unconscio-us,";
8097 PRINT "so you must land the
  aircraft alone. To be honest ";
8100 PRINT "he was not doing a
  great job before he pas-sed out.
  "
8105 PRINT : PRINT "You are in l
  evel flight at an altitude of
  3000m but only 20 km from the end
  of the runway. Your aircraft is
  pointing in ";
8110 PRINT "the cor-rect directi
  on but any crosswind will cause a
  deviation from this path."
8120 GO SUB 310
8130 PRINT "You must be able to
  land with an airspeed of about 50

```

```

  m/s with no appreciable roll or
  pitch."
8135 PRINT "Roll must be less th
  an 3 degrees and pitch must be b
  etween 0 and 5 degrees. Since the
  runway is 2000m long";
8140 PRINT "the distance indica
  t-or must show less than this fi
  g-ure on touchdown.";
8145 PRINT "Similarly the runwa
  y is 100m wide so that if the d
  eviation is greater than 50m on
  touchdown you will miss the runwa
  y."
8150 PRINT : PRINT "After touchd
  own you must use the rudder to co
  rrect the direction of travel an
  d the brakes to stop before you r
  each the end of the runway";
8160 PRINT ", (this is indicated
  by a distance reading of 0).";
8165 GO SUB 310
8167 PRINT "If you fail to stop
  or correct your direction (which
  may not be 0 when landing in a
  crosswind) you will crash."
8170 PRINT
8180 PRINT "You can see the ligh
  ts of the runway and in the di
  stance the lights of the city y
  ou hope to avoid."
8190 PRINT "Your instruments giv
  e you the following informatio
  n:"
8195 PRINT "AIRSPEED Speed in m
  /s."
8197 PRINT "BEARING Direction
  in which the aircraft is pointing
  . 0 is straight ahead. Posi
  tive values are to the right."
8199 PRINT "PITCH Positive val
  ues when the nose is pointing up."

```

```

8200 PRINT "ROLL    Positive valu
es clockwiseRoll causes banking
which          changes the aircraft
's bearing."
8201 GO SUB 310
8205 PRINT "ALT    Height above g
round in m.""DISTANCE  from end
of runway in m."
8206 PRINT "DEV    Sideways devia
tion from    centre of runway."
8220 PRINT "In flight you have t
he followingcontrols:"
8230 PRINT "'F roll to left'"G
maintain roll at this level"
8231 PRINT "H roll to right'"M
nose up(increase pitch)'"K main
tain pitch at this level'"O nos
e down(decrease pitch)"
8235 PRINT "0-9 throttle control
""Each setting has a terminal
value of speed which is reach
ed eventually."
8237 PRINT "e.g. 5-150 m/s  2-50
m/s.""Landing is usually reache
d on    setting 2"
8240 GO SUB 310
8250 PRINT "After touchdown only
the          following have effec
t:"
8260 PRINT "E rudder left""R ru
dder right""B apply brakes""
8265 PRINT "N.B. All controls ha
ve a delayedeffect on a large ai
rcraft."
8267 PRINT "    In general they h
ave an      effect on the second
display    after pressing a key
."
8269 PRINT "          Good luck!!!
"
8270 INPUT "Do you want to see t
hese again? ";x$

```

```

8280 GO TO 8060
8290 LET ci=PI/180: LET c2=180/P
I
8300 LET fd=0
8310 LET c5=.5
8320 LET lf=0: REM Landing flag
8330 LET pz=3000: REM Altitude
8340 LET py=-2E4: REM Distance f
rom airfield
8350 LET px=0: REM Lateral devia
tion from glide path
8360 LET rl=0: REM Roll
8370 LET pt=0: REM Pitch
8380 LET as=150: REM Airspeed
8390 LET ad=0: REM Air direction
8400 GO SUB 400
8410 CLS
8415 LET vz=0
8420 PRINT "Please select wind s
peed and    direction. A directi
on of 0      means that the wind
is blowing  straight at you and
this gives  the easiest landing
since you donot need to alter ro
ll or        bearing."
8430 PRINT "The direction should
be between -90, (from left) to
90, (from  right). A wind speed
of 5 is a  light breeze, 30 is
a gale."
8440 PRINT "Wind speed(m/s)": IN
PUT xo
8450 PRINT "Wind direction(deg)"
: INPUT x1
8460 LET wy=-xo*COS (x1*ci)
8470 LET wx=-xo*SIN (x1*ci)
8480 LET gz=vz: LET gy=vy+wy: LE
T gx=vx+wx
8490 LET tc=5: REM Throttle cont
rol
8500 LET bc=0: REM Banking contr
ol

```

```

8510 LET nc=0: REM Nose up/down
control
8520 LET rc=0: REM Rudder contro
l
8525 DIM t(10)
8530 FOR i=1 TO 10: READ t(i): N
EXT i
8540 DATA 0,25,50,75,100,150,200
,250,275,300
8550 LET yt=20: REM Yaw toleranc
e
8560 LET rt=3: REM Roll toleranc
e
8570 LET tp=5: REM Pitch toleran
ce
8580 LET lr=2000: REM Runway len
gth
8590 LET wr=50: REM Runway width
8600 LET hd=3e4: REM Distance to
horizon
8610 CLS
8620 PRINT PAPER 1;"Airspeed B
earing Pitch Roll ";AT 4,0; P
APER 0;"
      ";AT 5,0;" .! ! : . . . . ! . . .
: . . . . . . . . . . ! . : "
8630 PRINT AT 2,0; PAPER 1;"Alt
      Dist      Dev      ";AT
6,0; PAPER 0;" . . . . . ! . . . . .
. . . . . : . . . . : ";AT 21,0;"
      "
8640 GO TO 1000
9000 REM ***Final values***
9010 PRINT
9020 PRINT "Speed=";as;"m/s"
9030 PRINT "Direction=";ad;"deg"
9040 PRINT "Distance from end of
runway=";ABS py;"m"
9050 PRINT "Distance from centre
of runway=";ABS px;"m"
9060 PRINT "Pitch=";pt;"deg"
9070 PRINT "Roll=";rl;"deg"

```

```

9080 IF INKEY$="" THEN GO TO 90
00
9090 STOP
9095 FOR x=1 TO 40: BEEP .1,x: N
EXT x: RETURN
9097 FOR x=40 TO 1 STEP -1: BEEP
.1,x: NEXT x: RETURN

```





DAVID PONTING  
Original program by Alan Green

## GENERAL DESCRIPTION

In this enthralling little arcade game, you must save the planet Earth from a group of alien spacecraft who are trying to colonise it. If you allow five alien spacecraft to land, then the Earth has been colonised and you have lost. This game makes particularly good use of the Spectrum's sound, colour and graphics facilities.

## DETAILED DESCRIPTION

Lines 5 Call introductory music subroutine.  
35 Set highscore to 0.  
40-60 Set up user-defined graphics to represent the alien spacecraft.  
70-80 Set score, initial missile base x and y.  
90-97 If instructions are required, print them.  
100-120 Print the screen.  
150-155 Select a random starting position, direction of flight and type for the next alien to attack.  
160 Print missile base.  
170 Determine plotting x co-ordinate of base.

180 Move base right or left.  
190 Move plotting cursor to base's new position.  
200 If you have fired, draw the missile track.  
205 Move alien according to its direction of travel.  
210 Move alien down; if it lands, check to see if you have lost.  
212-214 Move the missile base again.  
220 Repeat main program loop.  
230-250 Print the alien who has landed below the line, and if five have landed, you lose.  
260-290 See if your missile hits the alien.  
300-340 Print game over, see if player wants to try again.  
350-370 Data for user-defined graphics.  
380-510 Instructions.  
520-640 Introductory music and picture.  
1000-3000 Move alien according to direction of travel.

```

5 GO SUB 520
10 REM ***Alien Blaster***
20 REM ***@ Alan Green ***
30 REM ***      1982      ***
35 LET ss=0
40 FOR n=0 TO 7: READ j: POKE
USR "a"+n,j: NEXT n
50 FOR n=0 TO 7: READ j: POKE
USR "b"+n,j: NEXT n
60 FOR n=0 TO 7: READ j: POKE
USR "c"+n,j: NEXT n
70 LET a=10: LET s=0: LET q=1
80 LET z=31: LET y=120
90 PRINT AT 12,1;"Do you want
instructions (y/n)"
95 BEEP .1,20: LET a$=INKEY$:
IF a$="y" THEN GO TO 380
96 IF a$="n" THEN GO TO 100
97 GO TO 95
100 BORDER 1: PAPER 1: INK 6: C
LS
110 PRINT AT 1,0: BRIGHT 1: INK
3: "_____

```

```

_____" : AT 19,0; INK 6; "_____
_____
120 PRINT AT 0,2; BRIGHT 1; INK
5; "Score 0"; AT 0,16; "High "; ss
150 LET c=INT (RND*30)+1; LET b
=3
152 IF c<=5 THEN LET w=2000
153 IF c>5 AND c<26 THEN LET w
=3000
154 IF c>=26 THEN LET w=1000
155 LET ch=INT (RND*2)+1
160 PRINT AT 18,a; INK 6; " A ";
AT b-1,c-1; INK 4; " "; AT b,c;
INK 4; CHR$ (144+ch)
170 LET x=(a*8)+12
180 LET a=a+(INKEY$="2" AND a<=
28)-(INKEY$="1" AND a>=1)
190 PLOT x,z: PLOT OVER 1;x,z
200 IF INKEY$="0" THEN DRAW I
NK 5;0,y: BEEP .025,20: PLOT x,z
: DRAW OVER 1;0,y: GO TO 260
205 GO SUB w
210 LET b=b+1: BEEP .01,19-b: I
F b=19 THEN GO TO 230
212 PRINT AT 18,a; INK 6; " A "
214 LET a=a+(INKEY$="2" AND a<=
28)-(INKEY$="1" AND a>=1)
220 GO TO 160
230 BEEP .5,-20
240 PRINT AT 18,c-1; " ": PRIN
T AT 21,q*2; INK 7; CHR$ (144+ch)
: IF q=5 THEN GO TO 300
250 LET q=q+1: GO TO 150
260 IF x=(c*8)+4 THEN GO TO 28
0
270 GO TO 205
280 FOR n=-3 TO -1: PRINT AT b,
c; INK 1; INVERSE 1; CHR$ (144+ch
): BEEP .04,5-n: PRINT AT b,c; I
NK 2; CHR$ (144+ch): BEEP .04,b:
NEXT n
290 LET s=s+(b+(ch*2)): PRINT A

```

```

T 0,8; BRIGHT 1; INK 5;s: PRINT
AT b,c; " ": GO TO 150
300 PRINT AT 10,12; INK 7; "Game
Over": IF s>=ss THEN LET ss=s:
PRINT AT 0,21; BRIGHT 1; INK 5;
#5
305 GO SUB 600
310 PRINT AT 12,6; INK 6; "Press
y to play again"
320 IF INKEY$="y" THEN CLS : B
EEP .3,30: GO TO 70
330 IF INKEY$="n" THEN CLS : S
TOP
340 GO TO 320
350 DATA BIN 00000000,BIN 00011
000,BIN 00011000,BIN 10011001,BI
N 10011001,BIN 11111111,BIN 1000
0001,BIN 10000001
360 DATA BIN 00111100,BIN 01111
110,BIN 11011011,BIN 01111110,BI
N 01011010,BIN 10000001,BIN 0100
0010,BIN 00100100
370 DATA BIN 00111100,BIN 00100
100,BIN 00111100,BIN 01111110,BI
N 11111111,BIN 00100100,BIN 0100
0010,BIN 10000001
380 BORDER 2: PAPER 2: INK 7: C
LS
390 PRINT "ALIEN BLASTER": PRIN
T AT 0,0; OVER 1; "_____
400 PRINT AT 2,0; "The Aliens ar
e invading your"; AT 4,1; "system,
your only hope is to"; AT 6,1; "bl
ast them before they land"; AT 8,
1; "and colonise the Earth!"
410 PRINT AT 10,1; INVERSE 1; I
NK 7; FLASH 1; "WARNING..."
420 PRINT AT 12,1; "ONLY FIVE AL
IENS HAVE TO LAND"; AT 14,1; "TO T
AKE OVER !!!"
430 PRINT AT 21,1; "Press any ke
y for next page"

```

```

440 BEEP .1,16: BEEP .1,21: IF
INKEY$="" THEN GO TO 440
450 CLS
455 PRINT "ALIEN BLASTER": PRIN
T AT 0,0; OVER 1;"-----"
460 PRINT AT 2,0;"To move base
right press "; INVERSE 1;"'2'"
470 PRINT AT 4,0;"To move base
left press "; INVERSE 1;"'1'"
480 PRINT AT 6,0;"Press "; INVE
RSE 1;"'0'"; INVERSE 0;" to fire
": PRINT AT 9,0;"There are two d
ifferent aliens, one scores more
than the other."
485 PRINT AT 12,0;"The aliens a
re worth more the nearer they
are to the planet."
487 PRINT AT 15,5; INK 5;"B...R
aider": PRINT AT 17,5; INK 4;"C.
..Droid"
490 PRINT AT 21,1;"Press any ke
y to start."
500 BEEP .2,5: IF INKEY$<>"" TH
EN GO TO 100
510 GO TO 500
520 BORDER 2: PAPER 2: INK 7: C
LS
530 PRINT AT 10,9;"Alien Blaste
r"
540 FOR n=-10 TO 40
550 BEEP .1,40-n
560 NEXT n: PAUSE 2: BEEP .5,20
: BEEP 1,15: NEXT n
610 FOR n=20 TO 0 STEP -2
620 BEEP .1,n: NEXT n
630 PAUSE 3: BEEP .25,15: BEEP
.25,15
640 RETURN
1000 LET c=c-1: RETURN
2000 LET c=c+1: RETURN
3000 RETURN

```

# STOCKMARKET

TIM WARD  
Original program by Bob Chappell

## GENERAL DESCRIPTION

Become a 'wheeler-dealer' in this game of high finance. You start off with between £100 and £100000 capital and deal in the four metals:- gold, tin, zinc and lead. The computer will recommend the best time to buy shares in these commodities, but ultimately the choice is yours. You can buy or sell as many shares as your income allows, then the computer will inform you whether the share values went up, down or remained constant.

At any time during the game you may receive a newsflash. These can be helpful i.e. bonuses which increase your cash or unwelcome taxes which deplete your funds.

The game can end in three possible ways:-  
You can quit.  
You can become bankrupt.  
The market may collapse.

## DETAILED DESCRIPTION

Lines 5-30 Initialisation and player's name input.  
40-50 Cheat routine.  
70-120 Press (SPACE) to continue routine.  
150-180 Input and validate capital.  
190-260 User input options.  
270-410 Buy or sell shares routine.  
430-540 Market news (up, down or hold).  
560-580 Profit or loss?  
590-620 Market fails.  
680-730 Bank takeover.  
740-780 BANKRUPT!  
800-990 Newsflashes.



1000-1190 Profit and loss account printout.  
 1200-1340 Portfolio printout.  
 1350-1430 End of game messages.  
 1450-1610 Initialise all variables.  
 1620-2020 Instructions.  
 3000-3050 Newsflash 'tickertape' routine.

```

5 REM *** 22/7/83 ***
10 REM *** Stock market ***
11 REM *** Tim Ward ***
12 REM *** Based on a ***
13 REM *** a PET program ***
14 REM *** by Bob Chappel ***
15 PAPER 7: INK 0: CLS : POKE 23609,50
20 CLS : PRINT AT 3,7; INVERSE 1;" Sto
ck market ": PRINT AT 5,13;"by": PRINT A
T 7,9;" Tim Ward": PRINT : PRINT " E
nter your name please": INPUT c$
25 POKE 23658,8
30 GO TO 1450
40 LET v$(1 TO )=" STOP PRESS! Mr "+c$
+" was today fined £1000 for attempting
to defraud the exchange ": LET o=o+80+LE
N c$: LET b=b-1000: GO SUB 3000
50 PAUSE 150: CLS : GO TO 430
70 PRINT #1;z$;
71 POKE 23692,255
80 LET a$=INKEY$: IF a$="" THEN GO TO
80
90 IF a$=" " THEN RETURN
100 GO TO 80
110 PAUSE 100: RETURN
120 CLS : PRINT AT 0,8;CHR$ 18+CHR$ 1;w
$: PAUSE 75: RETURN
150 CLS : PRINT AT 5,9; INVERSE 1;" Sto
ck market "
160 PRINT AT 7,1;" How much do you wish
to bank ? (100-100000)"
165 INPUT b
170 LET b=INT (b): IF b<100 OR b>100000
THEN GO TO 150

```

```

180 LET xz=1: GO TO 430
190 PRINT AT 17,0;h$: FOR j=1 TO 6: PRI
NT j;". ";m$(j);" ";
191 IF j=4 THEN PRINT " ";
192 NEXT j
200 PRINT AT 19,14;" Enter selection ":
INPUT c: LET c=INT (c)
210 IF c<1 OR c>6 THEN GO TO 40
220 IF c=5 THEN GO TO 440
230 IF c=6 THEN LET tt=1: GO TO 1010
240 IF f(c)=1 THEN PRINT d$: GO TO 430
250 PRINT AT 20,14;" How many shares ":
INPUT s: LET s=INT (s)
260 IF s<1 THEN GO TO 40
270 PRINT AT 21,5;"Buy or Sell !,(B/S)"
: INPUT a$
280 IF a$<>"B" AND a$<>"S" THEN GO TO
270
290 LET r=FN a(r): IF r=0 THEN PRINT s
$: GO TO 430
300 LET p=v(c)*(5^(4-c)): LET t=s*p: IF
a$="B" THEN GO TO 370
310 IF s>c(c) THEN GO TO 40
320 LET b=b+t: LET c(c)=c(c)-s
330 LET j=s*p(c): IF j=t THEN PRINT ;
INVERSE 1;" No loss or proffit ": GO TO
360
340 IF j<t THEN PRINT ; INVERSE 1;" A
profit of £";t-j;: LET l(c)=l(c)+(t-j):
GO TO 360
350 IF j>t THEN PRINT INVERSE 1;"A lo
ss of £";j-t;: LET l(c)=l(c)-(j-t)
360 GO TO 390
370 LET p(c)=p: IF b<t THEN PRINT f$:
GO TO 430
380 LET b=b-t: LET tc=p(c)*c(c): LET c(
c)=c(c)+s: LET f(c)=INT ((tc+t)/c(c))
390 LET r=FN a(r): IF r>0 THEN GO TO 4
30
400 LET r=FN a(r): IF r<5 THEN GO TO 4
00
410 GO SUB 70: GO TO 810

```

```

420 REM **market news**
430 GO SUB 70
440 LET 11=0: FOR j=1 TO 5: LET r=FN a(
r): IF r>4 THEN LET f(j)=0
441 NEXT j
450 CLS : PRINT AT 0,9;n$: PRINT AT 3,2
4; INVERSE 1;" Price "
451 FOR j=1 TO 4
460 PRINT AT 3+(j*2),1;j;"... ";m$(j);:
IF f(j)=1 THEN PRINT ";";p$: GO TO 56
0
470 LET r=FN a(r): IF r=0 THEN GO TO 6
80
480 LET x=r-INT ((v(j)*.4)+.9): IF x=0
THEN PRINT "; hold ";:
490 LET op=v(j)*(5^(4-j)): LET v(j)=v(j
)+x
500 LET cp=v(j)*(5^(4-j)): IF x<=0 THEN
GO TO 520
510 PRINT " up ";INT (cp-op);: GO
TO 530
520 IF x<0 THEN PRINT "; down ";INT (o
p-cp);:
530 PRINT AT 3+(j*2),24;" £";INT (cp);
540 IF cp<a(j) THEN PRINT "*": GO TO 5
60
560 NEXT j: IF 11<0 THEN PRINT AT 13,5
; INVERSE 1;" A total loss of ";ABS 11
570 IF 11>0 THEN PRINT INVERSE 1;" A
total profit of ";11
580 LET 11=0: IF xz=1 THEN LET xz=0: G
O SUB 70: GO TO 190
590 PRINT : PRINT INVERSE 1;" Bank ";:
LET r=FN a(r): IF r=0 THEN GO TO 630
600 LET x=INT (r-(.2*i)): LET i=i+x
610 IF i>20 THEN LET i=20
620 PRINT INVERSE 1;" rate ";i;" %": G
O TO 670
630 LET r=FN a(r): IF r<>0 THEN GO TO
650
640 LET v$(o TO )=" Market fails !!! ":
LET o=o+19: LET 11=11-b: LET 1(5)=1(5)-

```

```

b: LET b=0: GO TO 660
650 PRINT p$: LET f(5)=1
660 IF 11<0 THEN LET v$(o TO )=" loss
of"+"£"+STR$(ABS 11): LET o=o+10+LEN (S
TR$(ABS 11))+1
670 GO SUB 70: GO TO 800
680 LET f(j)=1: LET r=FN a(r): IF r<7 T
HEN GO TO 740
690 LET v$(o TO )=" Bank takeover ": LE
T o=o+16: LET r=FN a(r)
700 IF r=0 THEN LET v$=p$: GO SUB 3000
: GO TO 560
710 LET v$(o TO )=" sell "+m$(j)+" at £
": LET o=o+12+LEN (m$(j))+1: LET p=21*(5
^(4-j)): LET d=(20+(r/10))/100: LET v$(o
TO )=STR$(INT (p*d)): LET o=o+LEN (STR
$(INT (p*d)))
720 LET t=INT (p*d*c(j)): LET b=b+t: LE
T t1=t-c(j)*p(j): LET 11=11+t1: LET 1(j)
=1(j)+t1
730 LET c(j)=0: LET v(j)=1: LET p(j)=0:
GO SUB 3000: GO TO 560
740 IF r<2 THEN PRINT " ";p$: LET f(j)
=1: GO TO 560
750 PRINT " BANKRUPT!": LET o=o+26
760 LET 11=11-(c(j)*p(j))
770 LET 1(j)=1(j)-(c(j)*p(j))
780 LET c(j)=0: LET p(j)=0: LET v(j)=1:
GO TO 560
790 REM ***flash***
800 LET 12=0: LET r=FN a(r): IF r<5 THE
N GO TO 1010
810 IF r>7 THEN GO TO 890
820 LET r=FN a(r): IF r>4 OR r=0 THEN
GO TO 870
830 LET j=r: LET v$(o TO )=m$(j)+" bonu
s ": LET o=o+LEN (m$(j))+7: LET r=FN a(r
)
840 IF r=0 THEN LET v$(o TO )=p$: LET
o=o+LEN (p$)
850 LET r=10*r: LET v$(o TO )=STR$ r+"
% ": LET o=o+LEN (STR$ r)+3: LET cp=v(j)

```

```

*(5^(4-j)): LET 12=INT (cp/100*r)*c(j):
GO SUB 3000
860 LET b=b+12: LET 1(j)=1(j)+12: GO TO
970
870 LET v$(o TO )=" Tax bonus ": LET o=
o+12: LET r=FN a(r): IF r=0 OR b<1 THEN
LET r=0: GO TO 840
880 LET r=10*r: LET v$(o TO )=STR$ r+"
% ": LET o=o+LEN (STR$ r)+4: LET 12=INT
(b/100*r): LET b=b+12: LET 1(6)=1(6)+12:
GO TO 970
890 LET r=FN a(r): IF r=0 THEN GO TO 1
360
900 IF r<5 THEN GO TO 940
910 LET v$(o TO )=" Super tax ": LET o=
o+12: LET r=FN a(r): IF r=0 THEN GO TO
840
920 LET r=r*10: LET v$(o TO )=STR$ (r)+
" % ": LET o=o+LEN (STR$ r)+4: LET t=INT
(b/100*r): LET 1(6)=1(6)-ABS t: LET b=b
-ABS t
930 LET 12=12-ABS t: GO TO 970
940 LET v$(o TO )=m$(r)+" Bonus issue "
: LET o=o+LEN (m$(r))+14: LET t=p(r)*c(r)
)
950 LET c(r)=c(r)+(INT (c(r)/2))
960 IF c(r)<>0 THEN LET p(r)=INT (t/c(
r))
970 IF 12<0 THEN LET v$(o TO )=" A los
s of £"+STR$ (ABS 12): LET o=o+13+LEN (S
TR$ (ABS 12))+1
980 IF 12>0 THEN LET v$(o TO )=" A pro
fit of £"+STR$ 12: LET o=o+15+LEN (STR$
12)+1
990 GO SUB 3000
1000 REM *** P/L account***
1010 GO SUB 1020: GO TO 1210
1020 CLS : IF tt=1 THEN PRJ' INVERSE
1;"Closing ";
1030 PRINT INVERSE 1;" Profit and loss
account": LET t=0: LET q=29: FOR j=1 TO 4

```

```

1040 PRINT TAB 4;m$(j);
1050 IF 1(j)<0 THEN PRINT INVERSE 1;"
£";ABS 1(j);" ": GO TO 1070
1060 PRINT " £";1(j)
1070 LET t=t+1(j): NEXT j: LET 12=0: IF
tt=1 THEN GO TO 1090
1080 IF f(5)<>1 THEN LET 12=INT (b/100*
i): LET b=b+12
1090 LET f(5)=0: LET 1(5)=1(5)+12: PRINT
: PRINT TAB 4;"Interest ";
1100 IF 1(5)<0 THEN PRINT INVERSE 1;"
£";ABS 1(5);" ": GO TO 1120
1110 PRINT " £";1(5)
1120 PRINT TAB 4;"Tax ";
1130 IF 1(6)<0 THEN PRINT " "; INV
ERSE 1;"£";ABS 1(6);" ": GO TO 1150
1140 PRINT " £";1(6)
1150 LET t=t+1(5)+1(6)
1160 PRINT TAB 4;"Total ";: IF t<0 THEN
PRINT " "; INVERSE 1;"£";ABS t;" ":
GO TO 1180
1170 PRINT " £";t
1180 PRINT : PRINT : PRINT TAB 4; INVERS
E 1;" Inverse = loss "
1190 GO SUB 70: RETURN
1200 REM ***portfolio***
1210 CLS : IF tt=1 THEN PRINT INVERSE
1;" Closing ";
1220 PRINT INVERSE 1;" Portfolio ": PRI
NT 1$: PRINT AT 4,14;" Original": PRINT
AT 4,24;" Current"
1230 PRINT AT 6,1;"Metal Shares Cost
Price": PRINT 1$
1240 FOR j=1 TO 4: PRINT AT 6+(j*2),1;m$
(j)
1250 PRINT AT 6+(2*j),8;" ";c(j): IF c(j)
=0 THEN GO TO 1270
1260 PRINT AT 6+(2*j),14;" £";p(j)
1270 LET q=36: LET cp=v(j)*(5^(4-j)): PR
INT AT 6+(2*j),24;"£";cp: PRINT : NEXT j
1280 PRINT AT 16,1;"Bank £";b
1290 IF b<0 THEN PRINT INVERSE 1;" ove

```



```

rdrawn "
1300 IF b<-999 AND c(1)=0 AND c(2)=0 AND
c(3)=0 AND c(4)=0 THEN GO TO 1330
1310 PRINT : GO SUB 70: IF c=6 THEN GO
TO 1430
1320 GO TO 190
1330 PRINT INVERSE 1;" You are bankrupt
and your new "
1340 PRINT INVERSE 1;" address is Queer
Street!!! ": STOP
1350 REM End
1360 CLS : PRINT INVERSE 1;" MARKET COL
LAPSE!!": GO SUB 110: PRINT INVERSE 1;"
Bank takeover ": GO SUB 110
1370 LET q=23: FOR j=1 TO 4: PRINT " ";j
;".";m$(j);" sold at ";
1380 LET r=FN a(r): LET p=21*(5^(4-j)):
LET t=(100-(10*r))/100: LET r=INT (p*t):
LET l2=c(j)*r
1390 LET b=b+12: LET t=c(j)*p(j)
1400 LET l(j)=l(j)+(12-t)
1410 PRINT " £";r: NEXT j
1420 PRINT INVERSE 1;" Closing balance
£";b;" ": GO SUB 70: LET tt=1: GO SUB 10
20
1430 PRINT INVERSE 1;" End of speculati
ons!!": STOP
1440 REM ***Initialise***
1450 DIM m$(6,4): DIM v$(200): RANDOMIZE
: LET h$=" Buying and selling ": LE
T i=5: LET tt=0: LET o=1: LET y=0
1451 DIM v(10): DIM p(10): DIM e(10): DI
M f(10): DIM c(10): DIM a(10): DIM l(10)
1452 LET op=0: LET cp=0
1453 LET r=RND*(INT (65536*PEEK 23674+25
6*PEEK 23673+PEEK 23762))
1454 IF r=0 THEN GO TO 1453
1460 FOR j=1 TO 6
1461 READ m$(j)
1462 DATA "Gold","Tin","Zinc","Lead","Pa
ss","Quit"
1463 NEXT j

```

```

1470 LET l$="-----
-----"
1480 LET t$="
"
1490 FOR j=1 TO 4: LET v(j)=12: LET p(j)
=0: NEXT j: FOR j=1 TO 5: LET f(j)=0: NE
XT j
1500 DEF FN a(r)=INT (10*(RND*1)): FOR j
=1 TO 4: LET c(j)=0: NEXT j
1510 REM
1520 LET a(1)=1500: LET a(2)=300: LET a(
3)=60: LET a(4)=12: FOR j=1 TO 4: LET p(
j)=a(j): LET e(j)=a(j): NEXT j
1530 LET z$="Press <SPACE> to continue
"
1540 LET p$="suspended "
1550 REM
1560 REM
1570 LET d$=" dealing "+p$
1580 LET s$=" market "+p$
1590 LET f$=" Inadequate funds "
1600 LET n$=" Market news *="
Recomended"
1610 LET w$="News flash"
1620 CLS : PRINT " Stock market
"
1630 PRINT : PRINT " Do you want instruc
tions Y/N": INPUT a$
1640 IF a$(1)<>"Y" THEN GO TO 150
1650 CLS : PRINT " Stock market
"
1660 PRINT "You are first asked how much
"
1670 PRINT "capital you want to start wi
th."
1680 PRINT "Dealing is in four metals:-"
1681 PRINT "gold,tin,zinc and lead."
1690 PRINT "Various reports will help yo
u "
1700 PRINT "control your speculations."
1710 PRINT "The game ends when you decid

```

```

e to quit;": PRINT "or if you";
1720 PRINT " get too deeply in debt,": P
RINT "or if the market collapses."
1730 GO SUB 70: CLS : PRINT INVERSE 1;"
Market news /explanations "
1740 PRINT INVERSE 1;" Up ";; PRINT "
:-Price has risen"
1750 PRINT INVERSE 1;" Down ";; PRINT "
:-Price has fallen "
1760 PRINT INVERSE 1;" Hold ";; PRINT "
:-Price unchanged": PRINT
1770 PRINT INVERSE 1;" Suspended ": PRI
NT "No dealing allowed": PRINT
1780 PRINT INVERSE 1;" Takeover ": PRIN
T "Shares automatically": PRINT "sold at
price shown": PRINT
1790 PRINT INVERSE 1;" Takeover suspend
ed ": PRINT "Narrow escape!": PRINT
1800 PRINT INVERSE 1;" Bankrupt ": PRIN
T "Shares forfeited": GO SUB 70
1810 CLS : PRINT INVERSE 1;" Bank
rules ": PRINT
1820 PRINT INVERSE 1;" Rate x% ";; PRIN
T " Bank account increased": PRINT "
by x%"
1830 PRINT " (If overdrawn, inte
rest;": PRINT " payable on ove
rdraft)"
1840 PRINT INVERSE 1;" Suspended ";; PR
INT " No interest paid"
1850 PRINT INVERSE 1;" Fails;": PRINT "
All money lost !": PRINT "
(New bank arises ": PRINT "
next turn)"
1870 PRINT INVERSE 1;" Profit and loss
account": PRINT " Shows net;":
PRINT " gains and losses"
1880 PRINT INVERSE 1;" Portfolio": PRIN
T " Shows shares held and
bank account"
1890 GO SUB 70: CLS : PRINT INVERSE 1;"
Newsflashes"

```

```

1900 PRINT INVERSE 1;"Tax bonus x%";: P
RINT " Bank balance c
redited"
1910 PRINT INVERSE 1;" Supertax x%";: P
RINT " Bank balance d
ebited"
1920 PRINT INVERSE 1;" Metal bonus x%":
PRINT " Bank balance
credited ";
1930 PRINT "with incre
ase in value": PRINT TAB 13;"of holdings
"
1940 PRINT INVERSE 1;"Metal bonus issue
": PRINT TAB 13;" Holdings increased ";
1950 PRINT TAB 14;"by 1 share for ": PR
INT TAB 14;"every 10 held"
1960 PRINT INVERSE 1;"The above items m
ay be suspended"
1970 PRINT INVERSE 1;"at the last momen
t."
1975 GO SUB 70
1976 CLS
1980 PRINT INVERSE 1;" Market fails ";;
PRINT " End of game": PRINT "-all";
1990 PRINT TAB 5;"holdings sold at curre
nt": PRINT TAB 15;"market price."
2000 GO SUB 70: CLS : PRINT " You ar
e now about to become"
2010 PRINT " very rich or very poor.
"
2020 PRINT " But remember it's only
a game !": GO SUB 70: GO TO 150
3000 GO SUB 120: PRINT AT 3,0;t$: PRINT
AT 4,0;t$: PRINT AT 5,0;t$
3010 LET n=32: LET k=0: LET u=1
3015 FOR f=1 TO o: LET v=f
3016 IF f>31 THEN LET u=u+1
3017 IF f>31 THEN LET v=31
3018 IF f>200 THEN GO TO 3040
3020 PRINT AT 4,n-v; INVERSE 1; BRIGHT 1
;v$(u TO f)
3021 BEEP .1,20

```

```

3040 NEXT f
3041 PAUSE 50
3042 LET j=4
3050 LET v$="": LET o=1: RETURN

```

# ROCKET ATTACK

TIM WARD

Original program by John Gooderson

## GENERAL DESCRIPTION

In this exciting arcade-style game set in the near future, you are the commander of Britain's defences against Russian nuclear missiles. Your job is to fire your missiles at the incoming Russian nuclear missiles and destroy them in mid-air before they can reach the power stations and power lines that you are defending. The longer you last, the harder the game gets.

## DETAILED DESCRIPTION

Lines 50 Print instructions.  
 100 Set background colour and clear screen.  
 110 Set up variables.  
 120 Set up user-defined graphics and arrays.  
 900 Print bases.  
 1000 Calculate number of bases randomly.  
 1005-1020 Set up bases' positions and print highscore.  
 1021 Increment time counter.  
 1025 Move missile(s) down screen.  
 1026-1038 Move + up, down, right or left then print it.  
 1039 Specify which base fire will come from.  
 1040-1043 Set plot position of first remaining base.  
 1046 If firing, draw your missile's trail.  
 1047 If missile misses, print a small explosion.  
 1048 If missile hits, print a large explosion, and remove the destroyed missile from play.  
 1049 Repeat loop from line 1026.  
 1059 Randomly decide if missiles will be multiple warhead.  
 1065 Create multiple warheads.



1066 If all missiles have been destroyed, speed the next lot up.  
 1067 If you have survived 10 attacks, speed the missiles up.  
 1068 If you have survived 15 attacks then move the missiles to starting positions.  
 1080 Repeat loop from line 1010.  
 1085 Missile has reached the ground so print an explosion.  
 1086 Check to see if the missile has hit a base.  
 1091 Find position of first remaining base from the left.  
 1095 If all bases are destroyed then end.  
 2000 Increase the number of missiles.  
 3000-3050 Another game routine.  
 3055 Update highscore.  
 3100 Play the game again.  
 4000-4030 Set up the user-defined graphics.  
 4100-4400 Set up the arrays.  
 5000-5890 Print the instructions.

```

50 GO TO 5000
100 PAPER 0: BORDER 7: CLS
110 LET KOUNT=0: LET D=3: LET P=3: LET TOP=0: LET W=0: LET U=0: LET V=0: LET TOT=0: LET S=59: LET GAG=0: LET X1=19: LET Y1=10: LET X2=X1: LET Y2=Y1
120 GO TO 4000
130 REM SET UP BASES AND ROCKET ATTACK
700 PRINT AT 21,0: INK 4;B#
1000 LET D=3: LET N=INT ((RND*INT (P))+.5): IF N=0 THEN GO TO 1000
1005 LET W=0
1006 PRINT AT 0,9: INVERSE 1: INK 1;"HIGH SCORE ";TOP
1010 FOR J=1 TO N: LET N1=INT (RND*30)+2: LET A$(N1)=CHR$ 144
1020 NEXT J
1021 LET KOUNT=KOUNT+1
1025 FOR J=2 TO 20: PRINT AT J,0: INK M;A$: PRINT AT J-1,0;"

```

```

"
1026 FOR G=1 TO D
1027 REM CHECK KEY PRESS
1030 IF INKEY$="Q" OR INKEY$="q" THEN LET X1=X1-1
1031 IF INKEY$="A" OR INKEY$="a" THEN LET X1=X1+1
1032 IF INKEY$="O" OR INKEY$="o" THEN LET Y1=Y1-1
1033 IF INKEY$="P" OR INKEY$="p" THEN LET Y1=Y1+1
1034 IF X1>=19 THEN LET X1=19
1035 IF X1<=2 THEN LET X1=2
1036 IF Y1>=31 THEN LET Y1=31
1037 IF Y1<=1 THEN LET Y1=1
1038 PRINT AT X1,Y1: OVER 1: INK 1;"+" : PAUSE 7: PRINT AT X2,Y2: " ": REM FIRE CONTROL CURSOR
1039 LET X2=X1: LET Y2=Y1: REM SET FIRE FROM FIRST REMAINING BASE
1040 IF Y1<11 AND B(8)=3 THEN LET S=59
1041 IF Y1>11 AND Y1<17 AND B(14)=3 THEN LET S=107
1042 IF Y1>17 AND Y1<23 AND B(20)=3 THEN LET S=155
1043 IF Y1>23 AND B(26)=3 THEN LET S=203
1044 REM FIRE
1046 IF INKEY$="0" THEN PLOT 0 OVER 1: INK 2;S,8: DRAW OVER 1: INK 2;(Y1*8)-S,(20-X1)*8: FOR L=1 TO 5: BEEP .01,25+L: NEXT L: PLOT OVER 1;S,8: DRAW OVER 1;(Y1*8)-S,(20-X1)*8: LET GAG=1
1047 IF A$(Y1+1)=CHR$ 32 AND GAG=1 THEN PRINT AT X1,Y1: INK 3;CHR$ 148: BEEP .1,11: PAUSE 5: PRINT AT X1,Y1:" ": LET GAG=0: REM PRINT MINOR EXPLOSION IF TARGET MISSED

```

```

1048 IF GAG=1 AND A$(Y1+1)<>CHR$
  32 AND X1=J THEN PRINT AT X1,Y
  1; INK 2; BRIGHT 1;CHR$ 148: BEE
  P .1,15: PRINT AT X1,Y1; INK 2;
  BRIGHT 1;CHR$ 149: BEEP .1,18: L
  ET U=U+100: LET W=W+1: LET A$(Y1
  +1)=" ": LET GAG=0
1049 LET GAG=0: NEXT G
1059 LET FLAG=INT (RND*20)+1: IF
  INT (FLAG)/7=FLAG/7 AND N>3 AND
  J<15 THEN GO TO 1065: REM GENE
  RATE MULTIPLE WARHEADS
1060 GO TO 1066
1065 LET TAG=INT (RND*26)+4: IF
  A$(TAG)<>CHR$ 32 THEN LET A$(TA
  G)=CHR$ 145: LET A$(TAG-2)=CHR$
  145: LET A$(TAG+2)=CHR$ 145: LET
  N=N+1
1066 IF W=N THEN LET D=1: REM S
  PEED UP IF ALL MISSILES DESTROYE
  D
1067 IF KOUNT>10 THEN LET D=1:
  REM SPEED UP IF SURVIVED 10+ ATT
  ACKS
1068 IF KOUNT>15 THEN LET A$=A$
  (2 TO )+A$(1): LET D=2: REM MOVE
  MISSILES IF MORE THAN 15 ATTACK
  S
1080 NEXT J
1083 PRINT AT 20,0;"
"
1084 REM DETECT IF MISSILE HITS
  TARGET
1085 FOR J=1 TO 31: IF A$(J)<>CH
  R$ 32 THEN PRINT AT 21,J-1; INK
  2; BRIGHT 1; OVER 1;CHR$ 148: B
  EEP .1,11: PAUSE 5: PRINT AT 21,
  J-1; INK 2; OVER 1;CHR$ 149: BEE
  P .1,-15: PAUSE 5: PRINT AT 21,J
  -1;" ": LET V=V+B(J)*10
1086 IF A$(J)<>CHR$ 32 AND B(J)=
  3 THEN LET TOT=TOT+1: LET B(J)=

```

```

0: REM DETECTS IF ALL BASES DEST
  ROYED
1090 NEXT J
1091 FOR H=1 TO 31: IF B(H)=3 TH
  EN LET Y1=H-1: LET X1=19: GO TO
  1095: REM DETECTS FIRST SURVIVI
  NG BASE
1092 NEXT H
1095 IF TOT=4 THEN GO TO 3000:
  REM END IF ALL BASES DESTROYED
1100 LET A$( TO )=" "
2000 LET P=P+.5: GO TO 1000: REM
  INCREASE NO OF MISSILES
3000 PRINT #1;AT 1,0;"PRESS ANY
  KEY FOR A REPLAY": IF INKEY$=""
  THEN GO TO 3000
3050 PRINT #1;AT 1,0;"
"
3055 IF TOP<U-V THEN LET TOP=U-
  V
3100 CLS : LET KOUNT=0: LET P=3:
  LET A$( TO )=" ": GO TO 900
4000 LET N=9: FOR J=1 TO N: READ
  A: FOR K=0 TO 7: READ B: POKE U
  SR CHR$ (A)+K,B: NEXT K: NEXT J
4010 DATA 144,24,255,126,60,24,2
  4,24,24,145,0,0,0,66,126,60,24,2
  4,146,16,16,16,16,56,124,254,16
4020 DATA 147,16,16,255,255,255,
  126,60,255,148,0,60,126,255,255,
  126,60,0,149,162,164,9,66,36,145
  ,66,35
4030 DATA 150,31,15,7,7,15,31,63
  ,255,151,248,240,224,224,240,248
  ,250,255,152,36,60,36,60,36,126,
  129,129
4100 DIM B$(32): DIM B(32): DIM
  A$(32): DIM D$(32): DIM D(32)
4200 LET C$=CHR$ 150+CHR$ 151+CH
  R$ 147: FOR J=1 TO 31 STEP 2: LE
  T B$(J TO )=CHR$ 152: LET B(J)=2
  : NEXT J

```

```

4300 FOR J=6 TO 24 STEP 6: LET B
$(J TO J+2)=C$: LET B(J)=1: LET
B(J+1)=1: LET B(J+2)=3: NEXT J
4400 GO TO 900
5000 PAPER 7: INK 0: CLS :
5100 PRINT AT 0,7: INVERSE 1: BR
IGHT 1;" MISSILE ATTACK": PRINT
5200 PRINT "YOU ARE THE COMMANDE
R OF FOUR ANTI-MISSILE LASERS.
YOU ": PRINT "ARE DEFENDING THE
POWER STATIONSAND POWER LINES F
OR THE SOUTH OF"
5300 PRINT "ENGLAND. YOU MAY DIR
ECT FIRE BY MOVING THE CURSOR.":
PRINT : PRINT " 1..Q=UP": PR
INT " 2..A=DOWN": PRINT "
3..P=RIGHT": PRINT " 4..O=LEF
T": PRINT " 5..0=FIRE": PRINT

5400 PRINT "WHEN ALL YOUR BASES
HAVE BEEN DESTROYED THE GAME E
NDS.": PRINT "YOU MAY PLAY AGAIN
BY PRESSING ANY KEY.YOUR SCORE
WILL STILL BE RETAINED"
5500 PRINT #1;AT 1,0;"PRESS ANY
KEY TO CONTINUE": IF INKEY$="" T
HEN GO TO 5500
5550 PRINT #1;AT 1,0;"
": CLS

5600 PRINT "SCORING IS CALCULATE
D FROM THE TOTAL NO OF MISSILES
DESTROYED -": PRINT "THE NO OF
YOUR BASES & STATIONS AND POWER
LINES DESTROYED BYTHE ENEMY"
5700 PRINT AT 8,0;"ARE YOU USING
[1] A B/W OR [2]A COLOUR MONITO
R ?"
5750 IF INKEY$="1" OR INKEY$="2"
THEN GO TO 5800
5755 GO TO 5750
5800 IF INKEY$="1" THEN LET O=7

```

```

: LET I=0: LET M=1
5850 IF INKEY$="2" THEN LET O=1
: LET I=7: LET M=6
5890 GO TO 100

```



# TRUTH

DAVID JONES  
Original program by M. Costello

## GENERAL DESCRIPTION

Truth is a game to test your ingenuity. The computer picks out playing cards according to a certain rule, and your challenge is to work out which rule the computer is following. If you manage to guess eight cards correctly according to the rule, the computer realises that you have divined its secret, and goes on to another game according to a different rule. In case you get too complacent, there are three levels of play to work up through.

## DETAILED DESCRIPTION

Lines 10-130 Initialises variables.

140-240 Prints game rules and asks for level of play.

260-631 Plays game till no cards left or player has made eight correct guesses!

640-910 Picks random rule for basic level.

920-1180 Picks random rule for intermediate level.

1190-1370 Picks random rule for advanced level.

1381-2340 Level 1,2,3 tests.

2350 Input key routine.

2360-2570 Prints cards on screen. Asks for guesses, and shows mistakes and correct answers.

2580 End of game. Reveals rules.

2590-2821 Validates input and prints messages to screen.

2830-2991 Player has made eight correct scores. Prints score and plays again.

3000-3200 Validates input.

3290-3400 Sets up user-defined graphics.

```
10 REM TRUTH BY MIKE COSTELLO
20 REM CONVERTED TO SPECTRUM
30 REM FROM TRS-80 ORIGINAL
40 REM BY DAVID JONES 1983
50 INK 0: PAPER 6: CLS : BORDE
R 6: INPUT "": POKE 23658,8: POK
E 23609,20
60 PRINT "THIS IS THE GAME OF
TRUTH." "" "" "PLEASE WAIT A MOMENT
."
```

```
70 GO SUB 3290
80 DIM S$(13,4,2): DIM H(13,4)
: DIM T$(52,5): DIM U$(52,5): DI
M I(13): DIM J(13): DIM M(4): DI
M V$(5,5): DIM W$(4,3)
90 GO SUB 2810
100 LET ML=3: LET C7X=0: LET C7
Y=1: LET CMX=0: LET CMY=6: RANDO
MIZE
```

```
110 LET F=0: LET F0=0: LET F1=0
: LET G=0: LET G0=0
```

```
120 LET CT=52: FOR A=1 TO 4: LE
T S$(1,A)="A ": LET S$(11,A)="J
": LET S$(12,A)="Q ": LET S$(13,
A)="K ": FOR B=2 TO 10: LET S$(B
,A)=STR$(B): NEXT B: NEXT A
```

```
130 FOR A=1 TO 4: FOR B=1 TO 13
: LET H(B,A)=B: NEXT B: NEXT A
```

```
140 CLS : PRINT "TRUTH" "" "This i
s a card game where your task i
s to discover the""computers se
cret rule. The""computer will t
ell you if your cards follow th
e rule or not."
```

```
150 PRINT "If you lay 8 cards w
ithout any mistakes then the co
mputer will think you have guess
ed its rule."
```

```
160 PRINT "" "If you make a mista
ke when you enter your cards yo
u can erase your input by press
ing SHIFT 7."
```

```

170 PRINT "To enter your card p
ress""S for B, C for C etc."
180 PRINT "Level of play:"
190 PRINT "1. Basic""2. Inte
rmediate""3. Advanced""Pleas
e press 1,2 or 3";
200 GO SUB 2350
210 IF Z$="1" THEN LET O$="Bas
ic": GO SUB 640: GO TO 260
220 IF Z$="2" THEN LET O$="Int
ermediate": GO SUB 920: GO TO 26
0
230 IF Z$="3" THEN LET O$="Adv
anced": GO SUB 1190: GO TO 260
240 GO TO 200
250 REM RETURN HERE WITH RULE
260 CLS : GO SUB 2360: GO SUB 2
370: LET C7=64: LET CM=320: LET
C5=0: LET C6=0: LET K=0
270 LET ML=3: LET C=949: GO SUB
3000
280 IF Z$="Z" THEN GO TO 3200
290 GO SUB 2600: IF CJ=1 THEN
GO TO 270
300 GO SUB 350
310 IF CT=0 THEN GO TO 2580
320 IF K=8 THEN GO TO 2830
330 FOR A=1 TO 3: LET M(A)=M(A+
1): NEXT A: LET M(4)=JT: GO TO 2
70
340 REM THE FOLLOWING ARE ALL S
UBROUTINES
350 IF CT=52 THEN LET E=1: GO
TO 600: REM THE FIRST CARD IS AL
WAYS RIGHT
360 IF GC=1 THEN GO SUB 1350:
GO TO 600
370 IF GC=2 THEN GO SUB 1380:
GO TO 600
380 IF GC=3 THEN GO SUB 1410:
GO TO 600
390 IF GC=4 THEN GO SUB 1490:

```

```

GO TO 600
400 IF GC=5 THEN GO SUB 1510:
GO TO 600
410 IF GC=6 THEN GO SUB 1570:
GO TO 600
420 IF GC=7 THEN GO SUB 1610:
GO TO 600
430 IF GC=8 THEN GO SUB 1650:
GO TO 600
440 IF GC=9 THEN GO SUB 1690:
GO TO 600
450 IF GC=10 THEN GO SUB 1740:
GO TO 600
460 IF GC=11 THEN GO SUB 1770:
GO TO 600
470 IF GC=12 THEN GO SUB 1820:
GO TO 600
480 IF GC=13 THEN GO SUB 1860:
GO TO 600
490 IF GC=14 THEN GO SUB 1900:
GO TO 600
500 IF GC=15 THEN GO SUB 1940:
GO TO 600
510 IF GC=16 THEN GO SUB 1960:
GO TO 600
520 IF GC=17 THEN GO SUB 2000:
GO TO 600
530 IF GC=18 THEN GO SUB 2030:
GO TO 600
540 IF GC=19 THEN GO SUB 2080:
GO TO 600
550 IF GC=20 THEN GO SUB 2140:
GO TO 600
560 IF GC=21 THEN GO SUB 2180:
GO TO 600
570 IF GC=22 THEN GO SUB 2220:
GO TO 600
580 IF GC=23 THEN GO SUB 2290:
GO TO 600
590 IF GC=24 THEN GO SUB 2320
600 IF E=1 THEN GO SUB 2420: G
O TO 620

```

```

610 GO SUB 2500
620 LET CT=CT-1: IF CT=0 THEN
GO TO 2580
630 RETURN
640 GO SUB 1340
650 LET GC=INT (RND*8)+1: IF GC
>8 OR GC<1 THEN GO TO 650
660 IF GC=1 THEN GO SUB 740: R
ETURN
670 IF GC=2 THEN GO SUB 750: R
ETURN
680 IF GC=3 THEN GO SUB 760: R
ETURN
690 IF GC=4 THEN GO SUB 870: R
ETURN
700 IF GC=5 THEN GO SUB 880: R
ETURN
710 IF GC=6 THEN GO SUB 890: R
ETURN
720 IF GC=7 THEN GO SUB 900: R
ETURN
730 IF GC=8 THEN GO SUB 910: R
ETURN
740 LET F$="card": LET G$="colo
ur": LET H$="different from": L
ET I$=P$: LET J$="": LET K$="":
RETURN
750 LET F$="even": LET G$=P$: L
ET H$="odd": LET I$=Q$: RETURN
760 LET L=INT (RND*13)+1: IF L>
13 OR L<1 THEN GO TO 760
770 LET A$=S$(L,1): LET L1=L+6:
IF L1<14 THEN LET B$=S$(L1,1):
GO TO 790
780 LET B$=S$(L1-13,1): LET L1=
L1-13
790 IF L1=13 THEN LET C$=S$(1,
1): GO TO 810
800 LET C$=S$(L1+1,1)
810 LET L2=L1+6: IF L2<14 THEN
LET E$=S$(L2,1): GO TO 830
820 LET E$=S$(L2-13,1): LET L2=

```

```

L2-13
830 LET F$=A$+" to "+B$: LET G$
=P$: LET H$=C$+" to "+E$: LET I$
=Q$
840 FOR A=1 TO 13: LET I(A)=0:
LET J(A)=0: NEXT A: FOR A=1 TO 7
: LET I(A)=L: LET L=L+1: IF L=14
THEN LET L=1
850 NEXT A: FOR A=1 TO 6: LET J
(A)=L: LET L=L+1: IF L=14 THEN
LET L=1
860 NEXT A: RETURN
870 LET F$="card": LET G$="suit
": LET H$="consistent with seque
nce of: ": LET I$=W$(4)+" "+W$(2
)+" "+W$(1)+" "+W$(3)+" "+W$(4)+
" ....": RETURN
880 LET F$="card": LET G$="card
": LET H$="consistent with ": LE
T I$="sequence of: ": LET J$="2
odd alternating with 2 even (f
irst 2 always right)": RETURN
890 LET F$="card": LET G$="deno
mination": LET H$="1 to 3": LET
I$="greater than": LET J$=P$: R
ETURN
900 LET F$="card": LET G$="deno
mination": LET H$="1 to 3": LET
I$="less than": LET J$=P$: RETU
RN
910 LET F$="red": LET G$="previ
ous play": LET H$="mistake": RET
URN
920 GO SUB 1340
930 LET GC=INT (RND*10)+1: IF G
C>10 OR GC<1 THEN GO TO 930
940 IF GC=1 THEN GO SUB 1050:
GO TO 1040
950 IF GC=2 THEN GO SUB 1060:
GO TO 1040
960 IF GC=3 THEN GO SUB 1070:
GO TO 1040

```



```

970 IF GC=4 THEN GO SUB 1080:
GO TO 1040
980 IF GC=5 THEN GO SUB 1090:
GO TO 1040
990 IF GC=6 THEN GO SUB 1100:
GO TO 1040
1000 IF GC=7 THEN GO SUB 1110:
GO TO 1040
1010 IF GC=8 THEN GO SUB 1120:
GO TO 1040
1020 IF GC=9 THEN GO SUB 1170:
GO TO 1040
1030 IF GC=10 THEN GO SUB 1180
1040 LET GC=GC+8: RETURN
1050 LET F$="card": LET G$=F$: L
ET H$="consistent with ": LET I$
="sequence of: ": LET J$=W$(2)+"
"+W$(4)+" "+W$(1)+" "+W$(3)+" "
+W$(2)+" ....": RETURN
1060 LET F$="denomination": LET
G$=F$: LET H$="prime number": RE
TURN
1070 LET F$="suit": LET G$="it":
LET H$="consistent with ": LET
I$="this rule:"+CHR$ 13+"Divide
"+P$+" by 4,"+CHR$ 13+"if the re
mainder = 3 play "+W$(4)+CHR$ 13
+"if = 2 play "+W$(3)+CHR$ 13+"i
f = 1 play "+W$(2)+CHR$ 13+"else
play "+W$(1): RETURN
1080 LET F$="red": LET G$=P$: LE
T H$="even": LET I$=" and black
is correct if "+P$+" is odd": RE
TURN
1090 LET F$="card": LET G$="deno
mination": LET H$="4 to 6": LET
I$=" greater than": LET J$=P$: R
ETURN
1100 LET F$="card": LET G$="deno
mination": LET H$="8 to 10": LET
I$=" less than": LET J$=P$: RET
URN

```

```

1110 LET F$="card": LET G$="suit
or denomination": LET H$=" same
as": LET I$=P$: RETURN
1120 LET L5=INT (RND*4): IF L5>3
THEN GO TO 1120
1130 LET L6=L5: GO SUB 1140: LET
L7=L5: GO SUB 1140: LET L8=L5:
GO SUB 1140: LET L9=L5: LET L5=L
6: LET V$(2)=W$(L5+1): LET L5=L7
: LET V$(3)=W$(L5+1): LET L5=L8:
LET V$(4)=W$(L5+1): LET L5=L9:
LET V$(5)=W$(L5+1): GO TO 1160
1140 LET L5=L5+1: IF L5>3 THEN
LET L5=0
1150 RETURN
1160 LET F$="even denomination":
LET G$=P$: LET H$=V$(2)+" or "+
V$(3)+" and ": LET I$="odd denom
ination correct if "+P$+" is "+V
$(4)+" or "+V$(5): RETURN
1170 LET F$="black": LET G$="pre
vious play": LET H$="mistake": L
ET I$=" ": LET J$="": LET K$="re
d": LET L$=G$: LET M$="correct":
RETURN
1180 LET F$="card": LET G$="deno
mination": LET H$="within range
2 less than or 2 more than"+P$
: RETURN
1190 GO SUB 1340
1200 LET GC=INT (RND*6)+1: IF GC
>6 OR GC<1 THEN GO TO 1200
1210 IF GC=1 THEN GO SUB 1280:
GO TO 1270
1220 IF GC=2 THEN GO SUB 1290:
GO TO 1270
1230 IF GC=3 THEN GO SUB 1300:
GO TO 1270
1240 IF GC=4 THEN GO SUB 1310:
GO TO 1270
1250 IF GC=5 THEN GO SUB 1320:
GO TO 1270

```

```

1260 IF GC=6 THEN GO SUB 1330
1270 LET GC=GC+18: RETURN
1280 LET F$="even": LET G$=P$: LET H$="same colour as": LET I$=P$+"but one": LET J$="": LET K$="odd": LET L$=P$: LET M$="different colour from": LET N$=P$+"but one": RETURN
1290 LET F$="A to 7": LET G$=P$+"but 2": LET H$="black": LET I$="": LET J$="": LET K$="8 to K": LET L$=P$+"but 2": LET M$="red": RETURN
1300 LET F$="even": LET G$="previous play but 3": LET H$="correct": LET I$="": LET J$="": LET K$="odd": LET L$=G$: LET M$="mistake": RETURN
1310 LET F$="every third card": LET G$="it": LET H$="consistent with sequence of: ": LET I$=W$(2)+" "+W$(1)+" "+W$(4)+" "+W$(3)+" "+W$(2)+" ....": LET J$="(other cards are always right)": RETURN
1320 LET F$="court card": LET G$=P$: LET H$="not "+F$: LET I$=Q$: RETURN
1330 LET F$="red": LET G$="total of mistakes": LET H$="greater than or equal to": LET I$="total of correct cards": LET J$="": LET K$="black": LET L$=I$: LET M$="greater than": LET N$=G$: RETURN
1340 LET F$="": LET G$="": LET H$="": LET I$="": LET J$="": LET K$="": LET L$="": LET M$="": LET N$="": RETURN
1350 LET E=0: IF (F=0 OR F=3) AND (HS=1 OR HS=2) THEN LET E=1: RETURN

```

```

1360 IF (F=1 OR F=2) AND (HS=0 OR HS=3) THEN LET E=1
1370 RETURN
1380 LET E=0: LET L=1: POKE USR "E",L: POKE USR "E"+2,G: RANDOMIZE USR USUB: IF G=PEEK (USR "E"+2) THEN POKE USR "E",L: POKE USR "E"+2,HD: RANDOMIZE USR USUB: IF HD<>PEEK (USR "E"+2) THEN LET E=1: RETURN
1390 POKE USR "E",L: POKE USR "E"+2,HD: RANDOMIZE USR USUB: IF HD=PEEK (USR "E"+2) THEN LET E=1
1400 RETURN
1410 LET E=0: FOR A=1 TO 13: IF I(A)=G THEN GO TO 1430
1420 NEXT A: GO TO 1450
1430 FOR A=1 TO 13: IF J(A)=HD THEN LET E=1: RETURN
1440 NEXT A: RETURN
1450 FOR A=1 TO 13: IF J(A)=G THEN GO TO 1470
1460 NEXT A
1470 FOR A=1 TO 13: IF I(A)=HD THEN LET E=1: RETURN
1480 NEXT A: RETURN
1490 LET E=0: IF (F=3 AND HS=1) OR (F=1 AND HS=0) OR (F=0 AND HS=2) OR (F=2 AND HS=3) THEN LET E=1
1500 RETURN
1510 IF CT=51 THEN LET E=1: RETURN
1520 LET LC=1: LET L=1: POKE USR "E",L: POKE USR "E"+2,HD: RANDOMIZE USR USUB: IF HD=PEEK (USR "E"+2) THEN LET LC=0
1530 LET LF=1: POKE USR "E",L: POKE USR "E"+2,G: RANDOMIZE USR USUB: IF G=PEEK (USR "E"+2) THEN LET LF=0
1540 LET L0=1: POKE USR "E",G0:

```

```

POKE USR "E"+2,L: RANDOMIZE USR
USUB: IF G0=PEEK (USR "E"+2) THE
N LET L0=0
1550 LET E=0: IF (LC=1 AND LF=0
AND L0=0) OR (LC=1 AND LF=1 AND
L0=0) OR (LC=0 AND LF=1 AND L0=1
) OR (LC=0 AND LF=0 AND L0=1) TH
EN LET E=1
1560 RETURN
1570 LET GT=G: IF G=13 THEN LET
GT=0
1580 LET E=0: FOR A=1 TO 13: LET
I(A)=0: NEXT A: FOR A=1 TO 3: L
ET I(A)=GT+1: LET GT=GT+1: IF GT
=13 THEN LET GT=0
1590 NEXT A: FOR A=1 TO 3: IF I(
A)=HD THEN LET E=1: RETURN
1600 NEXT A: RETURN
1610 LET GT=G: IF G=1 THEN LET
GT=14
1620 LET E=0: FOR A=1 TO 13: LET
I(A)=0: NEXT A: FOR A=1 TO 3: L
ET I(A)=GT-1: LET GT=GT-1: IF GT
=1 THEN LET GT=14
1630 NEXT A: FOR A=1 TO 3: IF I(
A)=HD THEN LET E=1: RETURN
1640 NEXT A: RETURN
1650 LET E=0: IF HS=0 OR HS=3 TH
EN LET E=1: RETURN
1660 IF M(4)=1 THEN LET E=1
1670 RETURN
1680 REM END OF LEVEL 1 TESTS
1690 LET E=0: IF (F=1 AND HS=3)
THEN LET E=1: RETURN
1700 IF (F=3 AND HS=0) THEN LET
E=1: RETURN
1710 IF (F=0 AND HS=2) THEN LET
E=1: RETURN
1720 IF (F=2 AND HS=1) THEN LET
E=1
1730 RETURN
1740 FOR A=1 TO 13: LET I(A)=0:

```

```

NEXT A: FOR A=1 TO 3: LET I(A)=A
+1: NEXT A: LET I(4)=5: LET I(5)
=7: LET I(6)=11: LET I(7)=13
1750 FOR A=1 TO 7: IF HD=I(A) TH
EN LET E=1: RETURN
1760 NEXT A: LET E=0: RETURN
1770 IF (G=6 OR G=10) AND (HS=2)
THEN LET E=1: RETURN
1780 IF (G=5 OR G=9 OR G=13) AND
(HS=1) THEN LET E=1: RETURN
1790 IF (G=7 OR G=11) AND (HS=3)
THEN LET E=1: RETURN
1800 IF (G<5 OR G=8 OR G=12) AND
(HS=0) THEN LET E=1: RETURN
1810 LET E=0: RETURN
1820 LET L5=0: LET L=1: POKE USR
"E",G: POKE USR "E"+2,L: RANDOM
IZE USR USUB: IF G=PEEK (USR "E"
+2) THEN LET L5=1
1830 IF (HS=1 OR HS=2) AND (L5=0
) THEN LET E=1: RETURN
1840 IF (HS=0 OR HS=3) AND (L5=1
) THEN LET E=1: RETURN
1850 LET E=0: RETURN
1860 FOR A=1 TO 13: LET I(A)=0:
NEXT A: LET L1=G+4: IF L1>13 THE
N LET L1=L1-13
1870 FOR A=1 TO 3: LET I(A)=L1:
LET L1=L1+1: IF L1>13 THEN LET
L1=1
1880 NEXT A: FOR A=1 TO 3: IF HD
=I(A) THEN LET E=1: RETURN
1890 NEXT A: LET E=0: RETURN
1900 FOR A=1 TO 13: LET I(A)=0:
NEXT A: LET L1=G-8: IF L1<1 THEN
LET L1=L1+13
1910 FOR A=1 TO 3: LET I(A)=L1:
LET L1=L1-1: IF L1<1 THEN LET L
1=13
1920 NEXT A: FOR A=1 TO 3: IF HD
=I(A) THEN LET E=1: RETURN
1930 NEXT A: LET E=0: RETURN

```



```

1940 IF HS=F OR HD=G THEN LET E
=1: RETURN
1950 LET E=0: RETURN
1960 LET L=1: POKE USR "E",L: PO
KE USR "E"+2,HD: RANDOMIZE USR U
SUB: IF HD=PEEK (USR "E"+2) THEN
LET L=0
1970 IF (L=0 AND F=L8) OR (L=0 A
ND F=L9) THEN LET E=1: RETURN
1980 IF (L=1 AND F=L6) OR (L=1 A
ND F=L7) THEN LET E=1: RETURN
1990 LET E=0: RETURN
2000 IF (HS=0 OR HS=3) AND M(4)=
1 THEN LET E=1: RETURN
2010 IF (HS=1 OR HS=2) AND M(4)=
2 THEN LET E=1: RETURN
2020 LET E=0: RETURN
2030 FOR A=1 TO 13: LET I(A)=0:
NEXT A: LET L1=G-2: FOR A=1 TO 5
: IF L1<1 THEN LET L1=L1+13
2040 LET I(A)=L1: LET L1=L1+1: I
F L1>13 THEN LET L1=1
2050 NEXT A: FOR A=1 TO 5: IF HD
=I(A) THEN LET E=1: RETURN
2060 NEXT A: LET E=0: RETURN
2070 REM END OF LEVEL 2 TESTS
2080 IF CT>50 THEN LET E=1: RET
URN
2090 LET L=1: IF (F=0 OR F=3) AN
D (F0=0 OR F0=3) THEN POKE USR
"E",L: POKE USR "E"+2,HD: RANDOM
IZE USR USUB: IF HD<>PEEK (USR "
E"+2) THEN LET E=1: RETURN
2100 IF (F=1 OR F=2) AND (F0=1 O
R F0=2) THEN POKE USR "E",L: PO
KE USR "E"+2,HD: RANDOMIZE USR U
SUB: IF HD<>PEEK (USR "E"+2) THE
N LET E=1: RETURN
2110 IF (F=0 OR F=3) AND (F0=1 O
R F0=2) THEN POKE USR "E",L: PO
KE USR "E"+2,HD: RANDOMIZE USR U
SUB: IF HD=PEEK (USR "E"+2) THEN

```

```

LET E=1: RETURN
2120 IF (F=1 OR F=2) AND (F0=0 O
R F0=3) THEN POKE USR "E",L: PO
KE USR "E"+2,HD: RANDOMIZE USR U
SUB: IF HD=PEEK (USR "E"+2) THEN
LET E=1: RETURN
2130 LET E=0: RETURN
2140 IF CT>49 THEN LET E=1: RET
URN
2150 IF (F1=0 OR F1=3) AND HD<8
THEN LET E=1: RETURN
2160 IF (F1=1 OR F1=2) AND HD>7
THEN LET E=1: RETURN
2170 LET E=0: RETURN
2180 IF CT>48 THEN LET E=1: RET
URN
2190 LET L=1: POKE USR "E",L: PO
KE USR "E"+2,HD: RANDOMIZE USR U
SUB: IF HD=PEEK (USR "E"+2) AND
M(1)=1 THEN LET E=1: RETURN
2200 POKE USR "E",L: POKE USR "E
"+2,HD: RANDOMIZE USR USUB: IF H
D<>PEEK (USR "E"+2) AND M(1)=2 T
HEN LET E=1: RETURN
2210 LET E=0: RETURN
2220 LET E=1: IF CT>49 THEN RET
URN
2230 IF C5<>3*INT (C5/3) THEN R
ETURN
2240 IF HS=0 AND F1=1 THEN RETU
RN
2250 IF HS=3 AND F1=0 THEN RETU
RN
2260 IF HS=2 AND F1=3 THEN RETU
RN
2270 IF HS=1 AND F1=2 THEN RETU
RN
2280 LET E=0: RETURN
2290 IF HD>10 AND G<11 THEN LET
E=1: RETURN
2300 IF HD<11 AND G>10 THEN LET
E=1: RETURN

```

```

2310 LET E=0: RETURN
2320 IF C6>=C5 AND (HS=1 OR HS=2) THEN LET E=1: RETURN
2330 IF C6<C5 AND (HS=0 OR HS=3) THEN LET E=1: RETURN
2340 LET E=0: RETURN
2350 PAUSE 0: LET Z$=INKEY$: RETURN
2360 PRINT "CORRECT" "" "" "" "MISTAKES" "" "" "" "YOUR HAND": PRINT AT 19,0;"Give suit and denomination of the card you want to play: ": PRINT #1;X$: RETURN
2370 PRINT AT 11,0;"": FOR B=1 TO 4: PRINT W$(B): FOR A=1 TO 13: PRINT S$(A,B):: IF S$(A,B)="10" THEN PRINT " ";
2380 NEXT A: PRINT : NEXT B: PRINT AT 20,27;"": RETURN
2390 PRINT AT 1,0;"": FOR A=1 TO 52: PRINT T$(A):: NEXT A: PRINT AT 5,0;"MISTAKES": RETURN
2400 PRINT AT 6,0;"": FOR A=1 TO 52: PRINT U$(A):: NEXT A: PRINT AT 10,0;"YOUR HAND": RETURN
2410 GO SUB 2370: RETURN
2420 INPUT "": PRINT #1; FLASH 1;R$;"correct *": PAUSE 50: INPUT "": PRINT #1;X$;
2430 PRINT AT C7Y,C7X;W$(HS+1);S$(HD,HS+1)(1 TO 2): IF C7X>28 THEN LET C7X=0: LET C7Y=C7Y+1: GO TO 2460
2440 IF HD=10 THEN LET C7X=C7X+4: GO TO 2460
2450 LET C7X=C7X+3
2460 LET Z$=S$(HD,HS+1)(1 TO 2): LET S$(HD,HS+1)="": PRINT AT HS*2+12,0;"": FOR A=1 TO 13: PRINT S$(A,HS+1):: IF A=10 THEN PRINT " ";
2470 NEXT A: PRINT AT 20,27;"";

```

```

2480 LET T$(C5+1)=W$(HS+1)+Z$: LET C5=C5+1: LET F1=F0: LET F0=F: LET G0=G: LET F=HS: LET G=HD: LET JT=2: IF CT=52 THEN RETURN
2490 LET K=K+1: RETURN
2500 INPUT "": PRINT #1; FLASH 1;R$;"a mistake *": PAUSE 50: INPUT "": PRINT #1;X$;
2510 PRINT AT CMY,CMX;W$(HS+1);S$(HD,HS+1)(1 TO 2): IF CMX>28 THEN LET CMX=0: LET CMY=CMY+1: GO TO 2540
2520 IF HD=10 THEN LET CMX=CMX+4: GO TO 2540
2530 LET CMX=CMX+3
2540 LET Z$=S$(HD,HS+1)(1 TO 2): LET S$(HD,HS+1)="": PRINT AT HS*2+12,0;"": FOR A=1 TO 13: PRINT S$(A,HS+1):: IF A=10 THEN PRINT " ";
2550 NEXT A: PRINT AT 20,27;"";
2560 LET U$(C6+1)=W$(HS+1)+Z$: LET C6=C6+1
2570 LET K=0: LET JT=1: RETURN
2580 CLS : PRINT "Sorry, you have played all of your cards so I must now reveal the rule:"
2590 GO SUB 2850: GO TO 2880
2600 LET CJ=0: LET Z$=Y$(1 TO 1): IF Z$="S" OR Z$="H" OR Z$="D" OR Z$="C" THEN GO TO 2620
2610 GO TO 2800
2620 IF Z$="S" THEN LET HS=0: GO TO 2660
2630 IF Z$="H" THEN LET HS=1: GO TO 2660
2640 IF Z$="D" THEN LET HS=2: GO TO 2660
2650 LET HS=3
2660 LET Z$=Y$(2 TO 2): IF Z$<>"1" THEN IF LEN Y$=3 THEN GO TO 2800

```

```

2670 IF Z$<>"1" AND LEN Y$=3 THE
N GO TO 2800
2680 IF Z$<>"1" THEN GO TO 2710
2690 IF LEN Y$<>3 THEN GO TO 28
00
2700 IF Y$(LEN Y$ TO LEN Y$)<>"0
" THEN GO TO 2800
2710 IF LEN Y$=3 THEN LET HD=10
: GO TO 2780
2720 IF Z$="A" THEN LET HD=1: G
O TO 2780
2730 IF Z$="J" THEN LET HD=11:
GO TO 2780
2740 IF Z$="Q" THEN LET HD=12:
GO TO 2780
2750 IF Z$="K" THEN LET HD=13:
GO TO 2780
2760 IF Z$<"2" OR Z$>"9" THEN G
O TO 2800
2770 LET HD=VAL Z$
2780 IF H(HD,HS+1)=0 THEN GO TO
3190: REM NOT HOLDING THAT CARD
2790 LET H(HD,HS+1)=0: RETURN
2800 INPUT "": PRINT #1; FLASH 1
;"There is no such card!!";: PAU
SE 100: INPUT "": PRINT #1;X$;:
LET CJ=1: GO TO 3000
2810 LET W$(1)=CHR$(16)+CHR$(0
)+"B": LET W$(2)=CHR$(16)+CHR$(
2)+"D": LET W$(3)=CHR$(16)+CHR
$(2)+"A": LET W$(4)=CHR$(16)+C
HR$(0)+"C": LET X$="To give up
press: Z "
2820 LET P$=" last correct card"
: LET Q$=" and vice versa": LET
R$="* That card is ": RETURN
2830 CLS : PRINT "Well done! Wit
h 8 corect cards in a row, you
have worked out the secret rul
e (unless you're just lucky). T
he rule is : "

```

```

2840 GO SUB 2850: GO SUB 2970: G
O TO 2880
2850 PRINT 'F$;" is correct only
if "'G$'"is ";H$;I$'J$
2860 IF K$>" THEN PRINT "and"
K$;" is correct only if "'L$;" is
"'M$;N$
2870 RETURN
2880 PRINT AT 20,0;"To play agai
n type: G""To see the cards aga
in type: D";
2890 GO SUB 2350: IF Z$="G" THEN
GO TO 3260
2900 IF Z$="D" THEN CLS : LET C
7X=0: LET C7Y=1: LET CMX=0: LET
CMY=6: GO SUB 2360: INPUT "": PR
INT AT 19,0;"
To play again type:
G ""To see the rule again
type: R";: GO SUB 2390: GO SUB
2400: GO SUB 2410: GO TO 2920
2910 GO TO 2890
2920 GO SUB 2350: IF Z$="G" THEN
GO TO 3260
2930 IF Z$<>"R" THEN GO TO 2920
2940 CLS : GO SUB 2850
2950 IF K<>8 THEN GO TO 2880
2960 GO SUB 2970: GO TO 2880
2970 LET L1=44-C6
2980 LET D1=((CT/43)+(L1/44))*50
2990 PRINT AT 15,10;"S C O R E"
"Cards in hand = ";CT;"Mistakes
= ";C6;"Final score: ";D1'(";"0$
;" game)": RETURN
3000 LET Y$=""
3010 LET Z$="": IF Y$="" THEN P
RINT AT 20,27;" "AT 20,27;"
";
3020 GO SUB 2350
3030 IF Z$="Z" THEN RETURN
3040 IF Z$="." AND LEN Y$<2 THEN
GO TO 3010

```



```

3050 IF Z$="." THEN RETURN
3060 IF Z$=CHR$ 8 AND Y$="" THEN
  GO TO 3010
3070 IF Z$=CHR$ 8 AND Y$<>"" THE
N LET Y$=Y$(1 TO LEN Y$-1): GO
TO 3110
3080 IF Z$=CHR$ 13 THEN GO SUB
3180: GO TO 3000
3090 IF Z$<CHR$ 32 THEN GO TO 3
000
3100 LET Y$=Y$+Z$
3110 IF Z$="S" THEN PRINT W$(1)
;" ";; GO TO 3160
3120 IF Z$="H" THEN PRINT W$(2)
;" ";; GO TO 3160
3130 IF Z$="D" THEN PRINT W$(3)
;" ";; GO TO 3160
3140 IF Z$="C" THEN PRINT W$(4)
;" ";; GO TO 3160
3150 PRINT Z$;
3160 IF LEN Y$<=ML THEN GO TO 3
010
3170 INPUT "": PRINT #1; FLASH 1
;"Too many characters";: PAUSE 1
00: INPUT "": PRINT #1;X$;; GO T
O 3000
3180 INPUT "": PRINT #1; FLASH 1
;"Press '.' to play - not ENTER"
;; PAUSE 100: INPUT "": PRINT #1
;X$;; RETURN
3190 PRINT #1; FLASH 1;"You are
not holding that card!";: PAUSE
100: INPUT "": PRINT #1;X$;; LET
CJ=1: RETURN
3200 CLS : PRINT "If you are sur
e you want to giveup the attempt
to guess the      secret rule, h
it ENTER to see   what the rule
is. Otherwise     press any othe
r key to return tothe game."
3210 GO SUB 2350: IF Z$=CHR$ 13
THEN CLS : PRINT "The secret ru

```

```

le is:-": GO SUB 2850: GO TO 288
0
3220 LET SS1=C7X: LET SS2=C7Y: L
ET SS3=CMX: LET SS4=CMY
3230 LET C7X=0: LET C7Y=1: LET C
MX=0: LET CMY=6
3240 CLS : GO SUB 2360: GO SUB 2
390: GO SUB 2400: GO SUB 2410
3250 LET C7X=SS1: LET C7Y=SS2: L
ET CMX=SS3: LET CMY=SS4: GO TO 2
70
3260 FOR A=1 TO 52: LET T$(A)="
: LET U$(A)="": NEXT A: FOR A=1
TO 13: LET I(A)=0: LET J(A)=0: N
EXT A: FOR A=1 TO 4: LET M(A)=0:
NEXT A: GO TO 100
3270 PRINT INK 2;"A  "; INK 0;
"B  "; INK 0;"C  "; INK 2;"D
"
3280 STOP
3290 FOR i=0 TO 31: READ byte
3300 POKE USR "A"+i,byte: NEXT i
3310 DATA 0,8,28,62,127,62,28,8
3320 DATA 0,8,28,62,62,42,8,28
3330 DATA 0,28,28,107,127,107,8,
28
3340 DATA 0,34,119,127,127,62,28
,8
3350 LET USUB=USR "F"
3360 FOR A=0 TO 20: READ byte
3370 POKE USUB+A,byte: NEXT A: R
ETURN
3380 DATA 245,197,229,42,123,92,
6,32,35,16
3390 DATA 253,126,35,35,70,176,1
19,225,193,241
3400 DATA 201
3410 RANDOMIZE USR 64000
9999 REM 10 10 10 9999

```



STEVE BROCKBANK  
Original program by D. Denholm

## GENERAL DESCRIPTION

Although seemingly borrowed straight from the arcades, this game relies more on strategy than on quick reactions. You travel around the track, eating dots and try to avoid the enemy car travelling in the opposite direction. The control keys are very simple ... 'O' moves you OUT by one lane and 'I' moves you IN one lane. When the car collides with you, your score is given for the game, depending upon the number of dots you have eaten.

## DETAILED DESCRIPTION

Lines 10-375 Initialisation.

380-440 Countdown to start of game.

470-575 Move enemy car.

580-745 Move player.

810-840 Player and enemy car collide.

850-910 Print the score and ask Another go?

940-1200 Print out the playing track.

2000-2130 Print the instructions.

3000-3560 Set up the user-defined graphics characters.

4000-4100 Subroutine command which allows the SCREEN\$ command to recognise user-defined graphic characters.

```

10 DIM p(4): DIM m(4)
15 POKE 23658,0: REM **lower c
ase**
20 LET score=0: LET total=0: L
ET dot=0: LET mmove=0: LET pmove
=0: LET plus=0
100 GO SUB 3000
250 GO SUB 2000
350 GO SUB 940
360 LET p(1)=3: LET p(2)=16: LE
T px=1: LET py=0
365 LET m(1)=3: LET m(2)=14: LE
T mx=-1: LET my=0
370 INK 1: PRINT AT p(1),p(2):"
P"
375 INK 2: PRINT AT m(1),m(2):"
M"
380 INVERSE 1: PRINT AT 21,8:"O
N YOUR MARKS ! "
390 PAUSE 50: BEEP .5,.5
400 PRINT AT 21,8:"      GET SET
! "
410 PAUSE 50: BEEP .5,1
420 PRINT AT 21,8:"      GO !
"
425 PAUSE 50: BEEP .5,2
430 INVERSE 0
440 PRINT AT 21,8:"
"
470 REM * move missile *
480 LET y=m(1)+my: LET x=m(2)+m
x
490 GO SUB 4000
495 IF sc=0 THEN GO TO 525
500 IF sc=1 AND mmove=1 THEN L
ET plus=1: GO TO 530
505 IF sc=1 THEN LET dot=1: GO
TO 560

```

```

510 IF sc=2 THEN LET temp=my:
LET my=-mx: LET mx=temp: LET mmove=0: GO TO 480
520 IF sc=9 THEN GO TO 810
525 IF mmove=0 THEN LET mmove=1: GO TO 560
530 LET m(3)=y: LET m(4)=x
535 LET y=m(1)+mx: LET x=m(2)+m
y: GO SUB 4000: IF sc=0 THEN GO
TO 730
545 LET y=m(1)-mx: LET x=m(2)-m
y: GO SUB 4000: IF sc=0 THEN GO
TO 730
550 LET y=m(3): LET x=m(4)
560 PRINT AT m(1),m(2);" "
562 IF dot=2 THEN INK 0: PRINT
AT m(1),m(2);"0": LET dot=0
564 IF plus=1 THEN LET dot=1:
LET plus=0
565 INK 2: PRINT AT y,x;"M"
567 BEEP .01,9
570 LET m(1)=y: LET m(2)=x
575 LET dot=dot+dot
580 REM * move player *
610 LET d$=CHR$ (PEEK (23560))
615 POKE 23560,0
620 LET y=p(1)+py: LET x=p(2)+p
x
630 GO SUB 4000
635 IF sc=0 THEN GO TO 662
637 IF sc=1 AND pmove=1 THEN L
ET plus=1: GO TO 665
640 IF sc=1 THEN GO TO 690
650 IF sc=2 THEN LET temp=py:
LET py=px: LET px=-temp: LET pmo
ve=0: GO TO 620
660 IF sc=9 THEN GO TO 810
662 IF pmove=0 THEN LET pmove=
1: GO TO 690
665 IF d$<>"i" AND d$<>"o" THEN
GO TO 690
670 LET p(3)=y: LET p(4)=x

```

```

672 IF d$="o" THEN GO TO 685
675 LET y=p(1)+px: LET x=p(2)-p
y: GO SUB 4000: IF sc=0 THEN LE
T y=y+px: LET x=x-py: GO TO 690
680 LET y=p(3): LET x=p(4): GO
TO 690
685 LET y=p(1)-px: LET x=p(2)+p
y: GO SUB 4000: IF sc=0 THEN LE
T y=y-px: LET x=x+py: GO TO 690
687 LET y=p(3): LET x=p(4)
690 PRINT AT p(1),p(2);" "
692 IF sc=1 OR plus=1 THEN LET
score=score+10: BEEP .02,-3: LE
T plus=0
695 INK 1: PRINT AT y,x;"P"
697 BEEP .01,0
700 LET p(1)=y: LET p(2)=x
710 IF score>1270 THEN LET tot
al=total+score: LET score=0: GO
SUB 970
720 GO TO 470
730 IF RND<.6 THEN GO TO 550
735 LET plus=0
740 IF RND<.5 THEN LET y=y+mx:
LET x=x+my: GO TO 560
745 LET y=y-mx: LET x=x-my: GO
TO 560
810 REM *** crash ***
820 PRINT AT p(1),p(2);" "
825 PRINT AT m(1),m(2);" "
830 FLASH 1: PRINT AT y,x;" ":
PRINT AT 11,9;" CRASH !!! ": F
LASH 0
840 LET bord=3: FOR n=60 TO -60
STEP -2: BEEP .01,n: LET bord=7
-bord: BORDER bord: NEXT n: BORD
ER 3
850 BORDER 5: INK 1: PAPER 7: C
LS
855 LET total=total+score
860 PRINT AT 2,5;"You scored ";
total;" points"

```



```

870 PRINT AT 5,5;"by hitting ";
INT (total/10);" targets"
880 PRINT AT 12,0;"Do you wish
to play again? (Y/N)"
890 LET y$=INKEY$: IF y$="" THE
N GO TO 890
895 IF y$="y" OR y$="Y" THEN R
UN
900 IF y$<>"n" AND y$<>"N" THEN
GO TO 890
910 CLS : PRINT AT 12,12;"GOODB
YE!": PAUSE 0: GO TO 910
940 REM *** track ***
945 REM * N.B. letters in follo
wing PRINT statements are in gra
phics mode *
950 INK 4: PRINT "I...in      RAC
E CHASE   O...out"
960 PRINT "-----
-----"
970 INK 0: PRINT AT 2,0;"CAAAA
AAAAAAAAAAAAAAAAAAAAAD"
980 PRINT AT 3,0;"B G G G G G G
G G G G G G G B"
990 PRINT AT 4,0;"BGCAAAAAAAAA
AA AAAAAAAAAAADGB"
1000 PRINT AT 5,0;"B B G G G G G
G G G G G G B B"
1010 PRINT AT 6,0;"BGBGCAAAAAA
AA AAAAAAAAAAADGBGB"
1020 PRINT AT 7,0;"B B B G G G G
G G G G G G B B B"
1030 PRINT AT 8,0;"BGBGBGCAAAAA
AA AAAAAAAAAAADGBGBGB"
1040 PRINT AT 9,0;"B B B B G G G
G G G G G B B B B"
1050 PRINT AT 10,0;"BGBGBGBGCAAA
AAAAAAAAAADGBGBGBGB"
1060 PRINT AT 11,0;"B          B RA
CE CHASE  B          B"
1070 PRINT AT 12,0;"BGBGBGBGEAAA
AAAAAAAAAAGBGBGBGB"

```

```

1080 PRINT AT 13,0;"B B B B G G
G G G G G G B B B B"
1090 PRINT AT 14,0;"BGBGBGEAAAA
AAA AAAAAAAGBGBGB"
1100 PRINT AT 15,0;"B B B G G G
G G G G G G B B B"
1110 PRINT AT 16,0;"BGBGEAAAAAA
AAA AAAAAAAGBGB"
1120 PRINT AT 17,0;"B B G G G G
G G G G G G B B"
1130 PRINT AT 18,0;"BGEAAAAAA
AAA AAAAAAAGB"
1140 PRINT AT 19,0;"B G G G G G
G G G G G G G B"
1150 PRINT AT 20,0;"EAAAAA
AAAAAAAAAAG"
1160 PAPER 3: FOR n=0 TO 21: PRI
NT AT n,31;" ": NEXT n: PAPER 6
1200 RETURN
2000 BORDER 3: PAPER 6: INK 0: C
LS
2005 INVERSE 1
2010 PRINT AT 0,11;"RACE CHASE"
2015 INVERSE 0
2020 PRINT : PRINT "In this game
, you (": INK 1: PRINT "P": IN
K 0: PRINT ") hurtle"
2030 PRINT "round a track trying
to hit as "
2040 PRINT "many targets (6) as
you can "
2050 PRINT "before you are hit b
y a missile"
2060 PRINT "(": INK 2: PRINT "
M": INK 0: PRINT ") which is hu
rtling round the"
2070 PRINT "track in the opposit
e direction"
2080 PRINT "from you."
2090 PRINT : PRINT "Key I will m
ove you in one lane"
2095 PRINT : PRINT "Key O will m

```

```

ove you out one lane"
2100 PRINT : PRINT "PRESS ANY KE
Y TO PLAY"
2110 IF INKEY$="" THEN GO TO 21
10
2120 CLS
2130 RETURN
3000 REM ** set up user-defined
graphics **
3100 FOR n=0 TO 7: READ row
3110 POKE USR "M"+n,row: NEXT n
3160 DATA 0,153,90,60,126,102,16
5,153
3200 FOR n=0 TO 7: READ row
3210 POKE USR "P"+n,row: NEXT n
3260 DATA 0,0,16,124,254,56,0,0
3300 FOR u=1 TO 7
3305 FOR n=0 TO 7: READ row
3310 POKE USR CHR$ (143+u)+n,row
: NEXT n
3320 NEXT u
3400 RETURN
3500 DATA 0,0,0,255,255,0,0,0
3510 DATA 24,24,24,24,24,24,24,2
4
3520 DATA 0,0,0,15,31,24,24,24
3530 DATA 0,0,0,224,240,24,24,24
3540 DATA 24,24,24,31,15,0,0,0
3550 DATA 24,24,24,240,224,0,0,0
3560 DATA 0,0,0,24,24,0,0,0
4000 REM * routine to identify
user-defined graphics using SCRE
EN$ *
4010 POKE 23606,80: POKE 23607,2
53
4020 LET x$=SCREEN$ (y,x)
4025 POKE 23606,0: POKE 23607,60
4030 IF x$="G" THEN LET sc=1:
GO TO 4100
4040 IF x$=" " THEN LET sc=0:
GO TO 4100
4050 IF x$="A" OR x$="B" THEN

```

```

LET sc=2: GO TO 4100
4070 IF x$="C" OR x$="D" OR x$="
E" OR x$="F" THEN LET sc=4: GO
TO 4100
4080 IF x$="P" OR x$="M" THEN
LET sc=9: GO TO 4100
4090 PRINT "Error": PAUSE 0
4100 RETURN

```

# SPEC-TREK

JEREMY HANNETT  
Original author unknown

## GENERAL DESCRIPTION

Take the helm of your starship in this hectic, interactive game based on the television series 'Star Trek'. You hardly have time to communicate with your colleagues before the Klingons are upon you! At your disposal are Photon Torpedoes and Phasers for destroying the aliens, and it is advisable to strengthen your shields before combat.

You can call up a Sensor Scan which will tell you of any UFOs nearby, while you repair damage from the last battle. At any time you may change warp speed and vector direction (usually towards the nearest Starbase) to replenish your supplies. But do not linger there too long, for the Klingons are never very far away. Your main concern is your energy level, which can only be increased by successfully docking with a Starbase.

## DETAILED DESCRIPTION

Lines 10-310 Set up user-defined graphic characters, print the title page and user input for instructions.  
500-610 Set starship bridge display.  
1000-1260 Set initial variable values.  
2000-2900 Main program loop.  
3000-3100 Check for user input.  
3200-3295 Out of energy - End of game - Print rating - Another go?  
3300-3349 Update on-screen instrumentation.  
3400-3460 Reset Klingons.  
3500-3530 Reset Starbase.  
3600-3610 Short pause subroutine.  
3700-3720 Clear lower screen.

3800-3840 Leave Starbase.  
4000-4025 Carry out emergency repairs.  
4030-4095 Carry out general repairs.  
4100-4420 Fire Photon Torpedoes.  
5000-5325 Print Starbase - print explosion if hit - restock supplies.  
5500-5590 Meteorite shower damages warp engines.  
5600-5630 Escape from Klingons.  
6000-7220 Fire Phasers - check for hit on Klingon.  
7300-7390 Draw graphic starship.  
7400-7599 Play theme tune.  
8000-8190 Change warp factor.  
8200-8590 Sensor scan.  
8600-8650 Position of nearest starbase.  
8800-8860 Change the vector direction.  
9000-9170 Klingon attack!!!  
9200-9700 Calculate the damage sustained.  
9750-9790 User input for the next page.  
9800-9956 Print the instructions.

```
10 REM *** Spectrum Spectrek
30 RANDOMIZE
40 REM *** SET UP USER GRAPHIC
S
50 RESTORE : FOR i=0 TO 15: RE
AD a: POKE USR "a"+i,a: NEXT i
60 REM *** Graphics Data
70 DATA 0,1,195,87,141,70,57,1
6,0,0,131,204,34,196,56,8
200 LET h$=" S p e c - T r e k
"
205 BORDER 1
210 LET x=1: LET y=1: LET a=138
: LET b=80: GO SUB 7300
250 PRINT AT 5,7:
260 FOR i=1 TO LEN h$: PRINT I
NVERSE 1;h$(i);
270 FOR j=1 TO 20: NEXT j: NEXT
i
290 PRINT AT 18,0;"Do you want
```



```

instructions ? (Y/N)"
300 LET a$=INKEY$: IF a$="y" OR
a$="Y" OR a$="n" OR a$="N" THEN
GO TO 310
304 GO SUB 7400
305 GO TO 300
310 IF a$="y" OR a$="Y" THEN G
O SUB 9800
500 REM *** Display ***
510 PAPER 7: INK 0: CLS
520 PRINT "
"
530 FOR i=1 TO 9: PRINT "
";
550 NEXT i
560 PRINT "          U.S.S. ENTERP
RISE          "
570 PRINT "Star Date";TAB 18;"E
nergy"
580 PRINT "Condition";TAB 18;"S
hield"
590 PRINT "Warp      Vector      Se
nsors"
600 PRINT "Phasers";TAB 18;"Tor
pedoes"
610 PRINT "Damage";TAB 18;"Scor
e"
999 REM *** INITIALISE ***
1000 LET s$="DOWN": LET c$="GREE
N": LET t$="READY": LET r$="OFF"
1010 LET condition=4
1020 LET stx1=1: LET sty1=1: LET
stx2=1: LET sty2=1
1040 LET p$=t$: LET e=5000: LET
w=1: LET d=0: LET date=1000
1090 LET e$="
": LET st1=0: LET s
t2=0
1130 LET ss=0: LET score=0
1160 LET sh=0
1180 LET z$="          "
1190 LET docked=0

```

```

1200 LET we=1
1210 LET wmax=9
1220 LET nt=10
1230 LET p=1: LET v=1: GO SUB 35
00: GO SUB 3400
1260 LET u$="          "
2000 REM *** MAIN LOOP ***
2005 BEEP .005,condition+(6*(con
dition=2))
2010 LET date=date+1
2020 LET e=e-w*w-sh*10-ss*5-15*(
d>0)-5
2030 IF e<=0 THEN LET e=0
2040 GO SUB 3300
2050 IF e=0 THEN GO TO 3200
2100 IF w=0 THEN GO TO 2220
2110 PRINT AT sty1,stx1: INVERSE
1;" "
2120 PRINT AT sty2,stx2: INVERSE
1;" "
2130 LET stx1=INT (RND*30+1): LE
T sty1=INT (RND*9+1)
2140 LET stx2=INT (RND*30+1): LE
T sty2=INT (RND*9+1)
2150 PRINT AT sty2,stx2: INVERSE
1;"*"
2160 PRINT AT sty1,stx1: INVERSE
1;"*"
2170 LET bd=bd+w
2180 IF bv=v THEN LET bd=bd-w*2
2200 IF w<8 OR kv<>v THEN GO TO
2220
2205 IF dk>3 THEN GO TO 2210
2206 LET dk=2
2207 GO SUB 5600
2210 LET dk=dk+2+(w=9)
2220 IF dk>20 THEN GO SUB 3400
2300 LET dk=dk-(dk>0)
2310 IF bd<1 AND NOT docked THEN
GO SUB 5000
2400 IF ss AND (dk<3) THEN GO S
UB 8200

```

```

2500 IF NOT dk THEN GO SUB 9000
2600 IF d=0 THEN GO TO 2700
2610 IF dk THEN LET d=d-10
2620 IF d<10 THEN GO SUB 4030
2630 GO SUB 3300
2700 GO SUB 3000
2800 IF (d>40) AND RND<(w*w/2000
+.001) THEN GO SUB 5500
2900 GO TO 2000
2999 REM *** CHECK KEYBOARD
3000 LET a$=INKEY$
3010 IF a$="" THEN RETURN
3020 IF a$="w" OR a$="W" THEN G
O SUB 8000
3030 IF a$="s" OR a$="S" THEN G
O SUB 8100
3040 IF a$="p" OR a$="P" THEN G
O SUB 7000
3050 IF a$="z" OR a$="Z" THEN G
O SUB 8500
3060 IF a$="b" OR a$="B" THEN G
O SUB 8600
3070 IF a$="v" OR a$="V" THEN G
O SUB 8800
3080 IF a$="t" OR a$="T" THEN G
O SUB 4100
3090 IF a$="e" OR a$="E" THEN G
O SUB 4000
3100 RETURN
3200 REM *** OUT OF ENERGY.
3205 FOR i=20 TO -20 STEP -1: BE
EP .01,i: NEXT i
3210 PRINT AT 18,0; FLASH 1;"Out
of Energy.Mission Terminated"
3220 LET a$="AN ABSOLUTE DISASTE
R !!!"
3230 IF score>5 THEN LET a$="TE
RRIBLE."
3240 IF score>20 THEN LET a$="Q
uite Good."
3250 IF score>40 THEN LET a$="V
ery Good."

```

```

3260 IF score>60 THEN LET a$="E
XCELLENT!"
3270 IF score>100 THEN LET a$="
Klingon Zapper Supreme!!!"
3280 PRINT " Your Performance is
rated as:-": PRINT TAB 16-(LEN
a$/2); FLASH 1;a$;
3285 PRINT AT 0,0; FLASH 1;"Woul
d you like another flight ? "
3286 LET a$=INKEY$; IF a$="y" OR
a$="Y" OR a$="n" OR a$="N" THEN
GO TO 3290
3288 GO TO 3286
3290 IF a$="y" OR a$="Y" THEN R
UN
3291 FOR i=20 TO -20 STEP -1: BE
EP .01,i: NEXT i
3295 STOP
3300 REM *** UPDATE INSTRUMENTS
3301 BEEP .005,condition+(6*(con
dition=2))
3310 PRINT AT 11,10;date: PRINT
AT 11,28;" ": PRINT AT 11,28;
E
3320 PRINT AT 12,10;" "; I
NK condition; INVERSE 1;AT 12,10
;c$
3325 PRINT AT 12,28;" ";AT 12
,28; INVERSE 1;s$
3330 PRINT AT 13,7;w;AT 13,16;v
3335 PRINT AT 13,28;" "; INVE
RSE 1;AT 13,28;r$
3340 PRINT AT 14,10;" ";AT
14,10;p$;AT 14,28;" ";AT 14,2
8;nt
3345 PRINT AT 15,10;" ";AT 1
5,10;d;AT 15,28;" ";AT 15,28;
score
3349 BORDER condition
3350 RETURN
3400 REM *** RESET KLINGONS
3410 LET dk=INT (RND*10+10)

```

```

3420 LET nk=INT (RND*3+1)
3430 LET kv=INT (RND*8+1)
3440 LET c$="GREEN": LET conditi
on=4
3445 GO SUB 3300
3450 IF ss THEN PRINT AT 17,0;e
$;AT 17,0;"No Klingons in Sensor
Range."
3460 RETURN
3500 REM *** RESET STARBASE
3510 LET bd=INT (RND*200+200)
3520 LET bv=INT (RND*8+1)
3530 RETURN
3600 REM *** PAUSE ROUTINE
3610 FOR t=1 TO 100: NEXT t: RET
URN
3700 REM *** CLEAR LINES 20,21
3710 GO SUB 3600
3720 PRINT AT 20,0;e$;AT 21,0;e$
: RETURN
3800 REM *** LEAVE STARBASE
3810 LET docked=0
3820 GO SUB 3500
3830 PRINT AT 2,12;u$;AT 3,13;u$
;AT 4,10;u$;AT 5,11;u$
3840 RETURN
4000 REM *** EMERGENCY REPAIRS
4005 IF e>d+350 THEN GO TO 4025
4010 PRINT AT 20,0;"Not enough e
nergy for repairs"
4015 GO SUB 3700
4020 RETURN
4025 LET e=e-350
4030 REM *** GENERAL REPAIRS
4035 IF wmax=9 THEN GO TO 4060
4040 LET wmax=9
4045 PRINT AT 18,0;"Warp Engines
Repaired.Max Warp=9"
4050 GO SUB 3600
4055 PRINT AT 18,0;e$
4060 LET we=1
4065 LET p=1: LET p$="READY": LE

```

```

T d=0
4080 IF r$="OUT" THEN LET r$="O
FF"
4082 IF s$<>"OUT" THEN GO TO 40
90
4084 LET s$="DOWN": LET sh=0
4090 GO SUB 3300
4092 PRINT AT 16,0;e$
4095 RETURN
4100 REM *** PHOTON TORPEDOES
4110 IF nt THEN GO TO 4140
4120 PRINT AT 20,0;"No Torpedoes
"
4125 GO SUB 3700: RETURN
4140 IF dk=0 THEN GO TO 4162
4150 PRINT AT 20,0;"Target out o
f Range."
4155 GO SUB 3700
4160 RETURN
4162 IF s$<>"UP" THEN GO TO 417
0
4164 PRINT AT 20,0;"Shield must
be down to fire"
4165 PRINT "Torpedoes."
4166 GO SUB 3700: RETURN
4170 CIRCLE OVER 1;128,130,15
4172 PLOT OVER 1;114,130: DRAW
OVER 1;28,0: PLOT OVER 1;128,1
16: DRAW OVER 1;0,28
4175 PRINT AT 20,0;"Torpedo syst
ems active."
4180 LET kx=15
4190 FOR i=1 TO 9: PRINT AT i,kx
; OVER 1;"AB"
4200 PRINT AT i,kx; OVER 1;"AB"
4250 NEXT i
4260 FOR i=0 TO 9
4261 BORDER INT (RND*8): BEEP .0
1,20-(i*6)
4263 NEXT i: BORDER condition
4270 LET nt=nt-1
4280 LET nk=nk-1

```



```

4281 LET score =score+1
4282 GO SUB 3300
4290 IF nt AND nk THEN GO TO 41
80
4300 CIRCLE OVER 1;128,130,15
4302 PLOT OVER 1;114,130: DRAW
OVER 1;28,0: PLOT OVER 1;128,1
16: DRAW OVER 1;0,28
4310 GO SUB 3300
4320 IF nk THEN GO TO 4400
4330 GO SUB 3400
4370 GO SUB 3700
4380 RETURN
4400 GO SUB 3700
4410 PRINT AT 20,0;"Torpedoes ou
t.";AT 21,0;nk;" Klingons Left."
4415 GO SUB 3700
4420 RETURN
5000 REM *** STARBASE
5010 LET w=0: GO SUB 3300
5020 PRINT AT 2,12;" "
5030 PRINT AT 3,13;" "
5040 PRINT AT 4,10;">=" "<"
5050 PRINT AT 5,11;" "
5060 IF dk>2 THEN GO TO 5300
5070 PRINT AT 20,0;"Danger. Klin
gon Attack.": LET c$="RED": LET
condition=2: GO SUB 3300: GO SUB
3600
5080 PRINT AT 2,12;u$;AT 3,13;u$
;AT 4,10;u$;AT 5,11;u$
5090 GO SUB 3500
5100 LET zx1=15: LET zy1=5
5110 LET zx2=15: LET zy2=5
5120 LET zx3=15: LET zy3=5
5130 LET zx4=15: LET zy4=5
5135 BRIGHT 1
5140 FOR i=1 TO 5
5150 PRINT INVERSE 1;AT zy1,zx1
;"\\\"
5160 PRINT INVERSE 1;AT zy2,zx2
;"///"

```

```

5170 PRINT INVERSE 1;AT zy3,zx3
;"///"
5180 PRINT INVERSE 1;AT zy4,zx4
;"\\\"
5185 BEEP .002,5-i
5190 PRINT INVERSE 1;AT zy1,zx1
;" "
5191 PRINT INVERSE 1;AT zy2,zx2
;" "
5192 PRINT INVERSE 1;AT zy3,zx3
;" "
5193 PRINT INVERSE 1;AT zy4,zx4
;" "
5200 LET zx1=zx1-1: LET zx2=zx2+
1: LET zx3=zx3-1: LET zx4=zx4+1
5210 LET zy1=zy1-1: LET zy2=zy2-
1: LET zy3=zy3+1: LET zy4=zy4+1
5220 BEEP .002,5-i
5230 NEXT i
5240 BRIGHT 0
5275 LET docked=0
5280 GO SUB 3500
5285 PRINT AT 20,0;e$
5290 RETURN
5300 LET e=5000
5305 LET docked=1
5310 LET nt=nt+2*(nt<8)+(nt=9)
5315 GO SUB 3300
5320 GO SUB 4030
5325 LET bd=0: RETURN
5500 REM *** METEORITE SHOWER
5501 FOR i=1 TO (RND*100)+50: BO
RDER INT (RND*8): BEEP .005,INT
(RND*20): NEXT i
5510 LET tv=INT (RND*(wmax/2)+1)
5520 LET wmax=wmax-tv
5525 IF wmax<1 THEN LET wmax=1
5530 PRINT AT 18,0;"Meteorite Sh
ower.Damage to Warp Engines. Max
Warp = ";wmax
5540 GO SUB 3700
5550 PRINT AT 18,0;e$;AT 19,0;e$

```

```

5560 IF w<=wmax THEN RETURN
5570 LET w=wmax
5580 GO SUB 3300
5590 RETURN
5600 REM *** ESCAPED KLINGONS
5610 LET c$="GREEN": LET condition=4
5620 GO SUB 3300
5625 IF ss THEN PRINT AT 17,0;e$;AT 17,0;"No Klingons in Sensor Range"
5630 RETURN
6000 REM *** PHASER ROUTINE
6010 FOR i=1 TO 10
6020 NEXT i
6025 FOR i=2 TO 9
6030 PRINT AT i,15; OVER 1;"AB"
6050 IF INKEY$<>"" AND e>50 THEN GO TO 6190
6060 PRINT AT i,15; OVER 1;"AB"
6090 NEXT i
6100 RETURN
6190 LET e=e-50
6191 BORDER condition-1
6193 FOR q=1 TO -20 STEP -2: BEEP .01,q: BORDER INT (RND*8): PAPER INT (RND*8): NEXT q
6194 BORDER condition: PAPER 7
6200 PRINT AT i,15; OVER 1;"AB"
6205 IF i=5 THEN GO TO 6300
6210 GO TO 6080
6300 REM *** HIT KLINGON
6305 LET nk=nk-1
6306 LET score=score+1
6308 GO SUB 3300
6310 RETURN
7000 REM *** PHASERS
7010 IF p THEN GO TO 7040
7020 PRINT AT 20,0;"PHASERS ARE OUT"
7025 GO SUB 3700
7030 RETURN

```

```

7040 IF NOT dk THEN GO TO 7070
7050 PRINT AT 20,0;"Target out of Phaser Range"
7055 GO SUB 3700
7060 RETURN
7070 IF e>50 THEN GO TO 7100
7080 PRINT AT 20,0;"No Energy for Phasers"
7085 GO SUB 3700
7090 RETURN
7100 CIRCLE OVER 1;128,130,15
7102 PLOT OVER 1;114,130: DRAW OVER 1;28,0: PLOT OVER 1;128,16: DRAW OVER 1;0,28
7105 LET p$="ARMED": GO SUB 3300: PRINT AT 20,0;"Phasers Armed and Ready."
7110 GO SUB 6000
7115 GO SUB 3300
7120 IF e<50 THEN GO TO 7200
7130 IF nk=0 THEN GO TO 7150
7140 GO TO 7110
7150 GO SUB 3400
7160 LET p$="READY": GO SUB 3300
7180 CIRCLE OVER 1;128,130,15
7182 PLOT OVER 1;114,130: DRAW OVER 1;28,0: PLOT OVER 1;128,16: DRAW OVER 1;0,28
7185 PRINT AT 20,0;e$
7190 RETURN
7200 PRINT AT 20,0;e$;AT 20,0;"Energy Banks Drained"
7210 PRINT nk;"Klingons Left"; INVERSE 1;AT 14,6;"OUT"; INVERSE 0;" "
7215 GO SUB 3700
7220 GO TO 7180
7300 REM *** DRAW ENTERPRISE
7310 PLOT a,b: DRAW 0,6*y: DRAW 39*x,0,-0.5*SGN (x*y): DRAW 46*x,-16*y,-0.5*SGN (x*y): DRAW -42*x,0,-0.5*SGN (x*y)

```

```

7315 DRAW -43*x,15*y,-0.5*SGN (x
*y): INK 8: PLOT a+31*x,b+3*y: D
RAW 23*x,-4*y,0.3*SGN (x*y)
7320 DRAW -23*x,4*y,2*SGN (x*y):
PLOT a+85*x,b-10*y: DRAW 0,-6*y
: DRAW -36*x,0,-0.4*SGN (x*y)
7325 DRAW -48*x,16*y,-0.4*SGN (x
*y): DRAW -6*x,-4*y,0.4*SGN (x*y
): DRAW 14*x,-7*y,0.1*SGN (x*y)
7330 DRAW 5*x,4*y,-0.1*SGN (x*y)
: PLOT a+22*x,b-22*y: DRAW 7*x,-
2*y: PLOT a+22*x,b-27*y
7335 DRAW 2*x,10*y,1.9*SGN (x*y)
: DRAW -3*x,-10*y,3.1*SGN (x*y):
PLOT a+13*x,b-26*y: DRAW 3*x,14
*y,3.1*SGN (x*y)
7340 DRAW -4*x,-14*y,1.3*SGN (x*
y): DRAW -23*x,7*y,-0.1*SGN (x*y
): DRAW -24*x,13*y,0.1*SGN (x*y)
7345 DRAW 3*x,8*y,-2.1*SGN (x*y)
: PLOT a+8*x,b-11*y: DRAW 8*x,-1
*y,-0.3*SGN (x*y): PLOT a-5*x,b-
4*y
7350 DRAW -12*x,1*y: PLOT a-5*x,
b-4*y: DRAW -12*x,1*y: DRAW 15*x
,19*y: DRAW -6*x,1*y: DRAW -11*x
,-19*y
7355 PLOT a-16*x,b-4*y: DRAW -14
*x,13*y: PLOT a-17*x,b-4*y: DRAW
-9*x,0,-1.2*SGN (x*y): DRAW -11
*x,14*y
7360 PLOT a-20*x,b+8*y: DRAW -69
*x,12*y: DRAW 5*x,10*y,-0.8*SGN
(x*y): DRAW 67*x,-13*y
7365 DRAW -2*x,-9*y,-2.5*SGN (x*
y): DRAW 0,10*y,-1.3*SGN (x*y):
DRAW 31*x,-6*y: DRAW 0,10*y,2.4*
SGN (x*y)
7370 DRAW -2*x,-9*y,2.4*SGN (x*y
): PLOT a+11*x,b+22*y: DRAW -62*
x,10*y: DRAW -5*x,-7*y,1.1*SGN (
x*y)

```

```

7375 DRAW -4*x,6*y: DRAW 8*x,1*y
: PLOT a-89*x,b+21*y: DRAW -4*x,
9*y: DRAW 8*x,0
7390 RETURN
7400 REM *** STARTREK TUNE
7410 READ a,b: IF a=0 THEN REST
ORE 7500: GO TO 7410
7420 FOR i=1 TO a*2: BEEP .01,b:
BEEP .01,b+12: BEEP .01,b-12: N
EXT i: RETURN
7500 DATA 1,0,2,1,4,8,1,6,1,5,1,
3,1,1,1,0,4,-1
7510 DATA 1,0,2,1,4,8,1,6,1,5,1,
3,1,1,1,0,4,-1
7520 DATA 1,-3,4,-2,1,0,1,1,1,3,
1,5,1,6,2,8,4,11
7530 DATA 1,10,1,8,4,-2,2,0,4,1
7599 DATA 0,0
8000 REM *** CHANGE WARP
8005 IF w THEN GO TO 8025
8010 PRINT AT 20,0;"Warp Drive o
ut of action"
8015 GO SUB 3700
8020 RETURN
8025 LET ow=w
8030 PRINT AT 20,0;"New Warp Fac
tor ?"
8035 LET w$=INKEY$: IF w$<"0" OR
w$>"9" THEN GO TO 8035
8036 LET w=VAL w$
8040 IF w>wmax THEN GO TO 8070
8045 GO SUB 3300: PRINT AT 20,0;
e$
8050 IF ow=0 AND docked THEN GO
SUB 3800
8060 RETURN
8070 PRINT AT 20,0;"The Engines
canna take it , Captain!!"
8075 GO SUB 3700
8080 GO TO 8030
8100 REM *** SHIELD
8105 IF s$="OUT" THEN GO TO 8170

```



```

8110 LET sh=NOT sh
8130 LET s$="DOWN"
8140 IF sh THEN LET s$="UP"
8150 GO SUB 3300
8160 RETURN
8170 PRINT AT 20,0;"Shield is out of action."
8180 GO SUB 3700
8190 RETURN
8200 REM *** SENSORS DETECT KLINGON
8205 IF dk<>2 THEN GO TO 8230
8210 LET c$="YELLOW": LET condition=6
8220 GO SUB 3300
8230 LET tv=kx+4
8240 IF tv>8 THEN LET tv=tv-8
8250 PRINT AT 17,0;nk;" UFOs, ";dk;" Light Years, Vector ";tv
8260 RETURN
8500 REM *** SENSORS
8510 IF r$<>"OUT" THEN GO TO 8540
8520 PRINT AT 20,0;"Sensors out of action."
8523 GO SUB 3700
8530 RETURN
8540 LET ss=NOT ss
8550 LET r$="OFF"
8560 IF ss THEN LET r$="ON"
8570 GO SUB 3300
8580 IF ss THEN PRINT AT 17,0;"No Klingons in Sensor Range."
8590 RETURN
8600 REM *** NEXT STARBASE
8620 PRINT AT 20,0;"Nearest Starbase ";bd;" Light"
8625 PRINT "Years in Vector ";bv
8630 GO SUB 3700
8650 RETURN
8800 REM *** CHANGE VECTOR
8810 PRINT AT 20,0;"New Vector ?"

```

```

8820 LET v$=INKEY$: IF v$<"1" OR v$>"8" THEN GO TO 8820
8830 LET v=VAL v$
8850 GO SUB 3300: PRINT AT 20,0;e$
8860 RETURN
9000 REM *** KLINGON ATTACK
9010 LET c$="RED": LET condition=2
9020 GO SUB 3300
9025 IF docked THEN GO SUB 5080
9030 LET kx=15: LET ky=1
9040 FOR i=1 TO 9
9050 PRINT INVERSE 1;AT ky,kx,"AB"
9060 PRINT INVERSE 1;AT ky,kx,""
9080 LET ky=ky+1
9100 NEXT i
9110 FOR i=1 TO nk*2
9120 BORDER INT (RND*8): BEEP .005,25
9140 NEXT i
9160 GO SUB 9200
9170 RETURN
9200 REM *** CALCULATE DAMAGE
9210 LET d=d+nk*(20+20*(NOT sh))
9215 GO SUB 3300
9220 IF d>400 THEN GO TO 9300
9230 LET tv=INT (d/100)
9240 GO TO 9700-100*tv
9300 LET we=0
9310 LET w=0
9320 GO SUB 3300: PRINT AT 16,0;"Warp engines "; INVERSE 1;"OUT"
9400 LET p=0
9410 LET p$="OUT"
9500 LET s$="OUT"
9510 LET sh=0
9600 LET ss=0: LET r$="OUT"
9620 PRINT AT 17,0;e$: GO SUB 3300

```

```

9700 RETURN
9750 REM *** NEXT PAGE
9760 PRINT AT 21,3; FLASH 1;" Press any Key to Continue."
9770 GO SUB 7400: IF INKEY$="" THEN GO TO 9770
9780 CLS
9790 RETURN
9800 REM *** INSTRUCTIONS
9802 CLS
9804 PRINT "You are in command of Starship Enterprise. Your mission is to destroy Klingon Battle Cruisers in this quadrant of the galaxy. You are armed with Phasers and Photon Torpedoes. Torpedoes can only be fired if your shield is down, and you have a limited supply, but they never miss."
9822 PRINT "Phasers are fired by you and sometimes miss. They use 50 units of energy per shot. If you run out of energy the mission ends. Energy drains are Life Support (cannot be switched off) Shield (if on), and Damage Repair";
9840 PRINT "Damage is sustained as a result of Klingon attack, and is worse if unshielded."
9846 GO SUB 9750
9847 PRINT "If damage exceeds 100"
9848 PRINT "units, sensors are lost; >200 - shield is lost; >300 - phasers are lost >400 - warp drive lost. Functions are not restored until less than 10 units, or you request emergency repairs (Uses a lot of energy), or until you dock at a starbase."

```

```

9864 PRINT "Each starbase can refuel only 1 ship, so you will need to call at several during a long mission, but don't lead the Klingons there, or the base will be destroyed."
9876 PRINT "Commands are given by pressing a certain key (you may have to wait a few seconds before the ship's computer obeys)."
9884 GO SUB 9750
9886 PRINT "Options are:-
          S - Change Shield Status
          Z - Change Sensor Status
          W - Change Warp Factor (0 TO 9)
          V - Change Vector (Direction 1-8)
          E - Do Emergency Repairs
          T - Fire Photon Torpedoes
          P - Arm Phasers. You then fire by pressing any key (Hold down for continuous fire)."
9906 PRINT "B - Gives distance and direction of next starbase.
          Vectors are :- 1
                        8 2
                        7 3
                        6 4
                        5"
9910 PRINT "At Warp 8 or 9 you can outrun Klingons from the opposite direction. eg. if they come from Vector 2, you must take Vector 6.": GO SUB 9750
9932 PRINT "The display shows the main screen and instrument panel of the ENTERPRISE. You must keep alert for attacks by Klingons (who look like AB) and try to conserve energy so as to reach the next starbase."

```

```

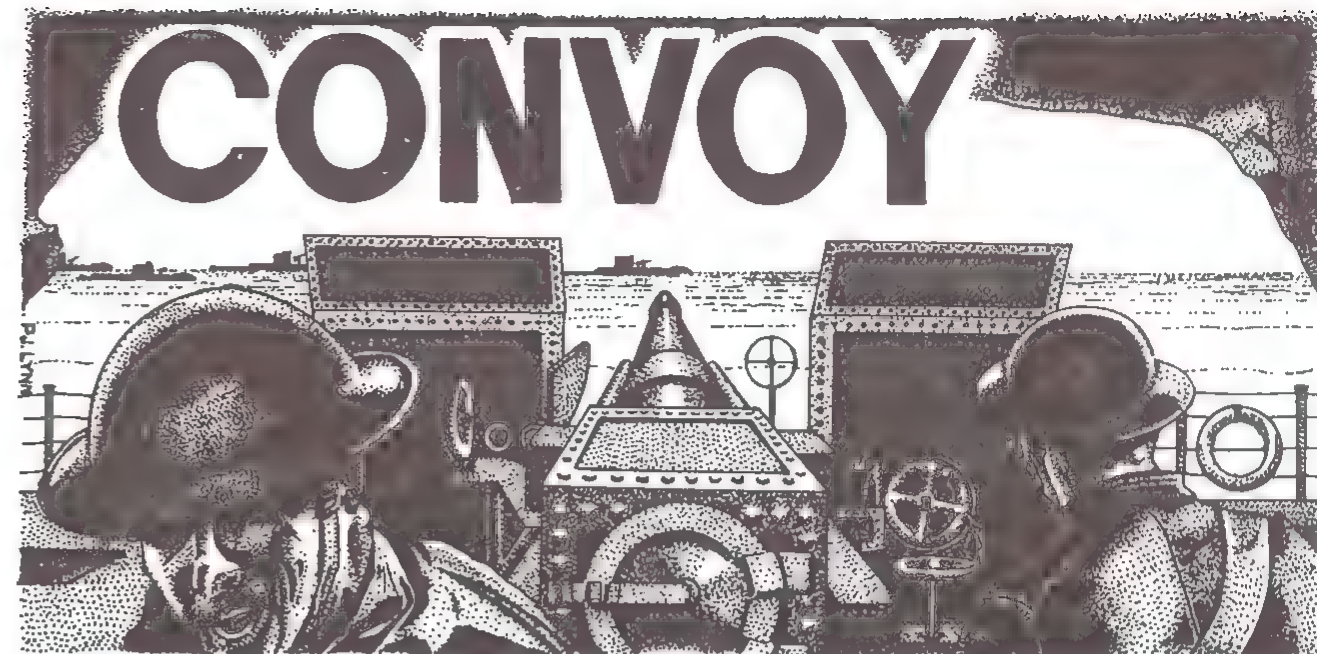
        If sensors are on, y
ou get a warning of klingons
nearby."
9950 PRINT : PRINT : PRINT FLAS
H 1;"      GOOD LUCK CAPTAIN !!
"

```

```

9955 LET x=-1: LET y=1: LET a=10
0: LET b=40: GO SUB 7300
9956 GO SUB 9750: RETURN
9999 GO SUB 5070: GO TO 9999

```



ANDREW ESMOND  
Original program by Jeff Aughton

---

## GENERAL DESCRIPTION

---

In this game you are the commander of a coastal gun and it is your job to sink the enemy convoy before it passes through the straits which your gun overlooks. You can control the elevation and bearing of your gun to enable you to hit the ships as they move through the channel.

This game uses the colour, sound and user-defined graphics of the Spectrum in a most effective manner and will provide a good example for you to base some of your own games on.

---

## DETAILED DESCRIPTION

---

Lines 10-60 Set up variables and arrays.

70-250 Print the picture of the straits.

260-270 Print the ships in their original positions.

280 Print gun elevation and bearing.

300-340 Get keyboard input and, if necessary, adjust elevation and bearing appropriately.

350-410 If a shell has been fired, determine where it lands.

420-470 Move the next ship in the convoy.

600-740 A shell has landed so simulate the explosion, and



if a ship has been hit, remove it from play.  
 800-860 You have destroyed the whole convoy, print number of ships sunk and duration of game, then stop.  
 880-910 Make the next game harder, if the player wants it that way.  
 920-960 A ship has escaped so you have lost.  
 980-990 User-defined graphics data.  
 1000-1080 Instructions routine.  
 9000-9030 Setting up user-defined graphics and a bit more user-defined graphics data.

```

10 REM convoy
11 LET t=0: LET ts=0
15 GO SUB 1000
20 LET n=3: LET u=.058: LET v=.04: LET
p=P1/180: LET l=23205
30 DIM m(14): DIM x(9): DIM y(9): DIM
s(9)
40 LET b=0: LET e=10: LET h=0: LET k=0
: LET r=0
60 RESTORE : FOR i=0 TO 13: READ m(i+1
): NEXT i
70 REM ***draw***
80 CLS : FOR i=1 TO 15: PRINT INVERSE
1; INK 4;" "": NEXT i: PRINT INVERSE
1; INK 4;" "
100 PRINT TAB 21; INK 4; INVERSE 1;"
"
110 PRINT TAB 23; INK 4; INVERSE 1;"
"
120 PRINT TAB 25; INK 4; INVERSE 1;"
"
130 PRINT TAB 27; INK 4; INVERSE 1;"
"
140 PRINT TAB 29; INK 4; INVERSE 1;"
"
150 PRINT AT 15,0; INK 4; INVERSE 1;"
"
160 PRINT INK 4; INVERSE 1;" "
170 PRINT INK 4; INVERSE 1;" "
180 PRINT INK 4; INVERSE 1;" "

```

```

190 PRINT INK 4; INVERSE 1;" "
200 PRINT INK 4; INVERSE 1;" "
210 PRINT INK 4; INVERSE 1;" +
";AT 20,11;
230 FOR i=0 TO 21
240 PRINT AT i,31; INK 4; INVERSE 1;" "
: NEXT i
250 FOR i=6 TO 25: PRINT AT 21,5+i; INK
4; INVERSE 1;" "": NEXT i
260 FOR i=1 TO n: READ x(i),y(i)
265 LET s(i)=22528+x(i)+(32*y(i))
270 PRINT AT y(i),x(i); INK 7;" "": NEX
T i: LET sb=FN x()
280 PRINT AT 0,0; INK 4; INVERSE 1;"hit
s";h;" elevation ";e;" bearing ";b;" "
300 LET k=k+1: IF k>n THEN LET k=1
320 LET m$=INKEY$
330 LET e=e-(e>10)*(m$="2")+(e<85)*(m$=
"8")
340 LET b=b-(b>-5)*(m$="4")+(b<90)*(m$=
"6")
350 IF m$<>"5" OR r<>0 THEN GO TO 420
370 FOR i=1 TO 7: LET ts=200*SIN (e*p):
LET t=FN x()
380 NEXT i
390 LET r=23*SIN (2*e*p)
400 LET x=INT (1.2*r*SIN (b*p)+.5)
410 LET y=INT (r*COS (b*p)+.5)
420 IF r<>0 AND RND>.85 THEN GO TO 600
430 LET dx=(RND<u): LET dy=(RND<v): LET
d=2*dx+64*dy
440 LET m=s(k)+d: IF PEEK m=4 THEN GO
TO 920
450 IF PEEK m<>9 OR PEEK (m+1)<>9 THEN
GO TO 280
460 POKE s(k),9: POKE s(k)+1,9: PRINT A
T y(k),x(k):" "": LET s(k)=m: LET x(k)=x
(k)+dx*2: LET y(k)=y(k)+2*dy
470 PRINT AT y(k),x(k); INK 7;" "
475 GO TO 280
600 REM ***shell lands***

```

```

610 LET r=0: LET w=3: LET q=1+x-32*y
620 IF PEEK q=15 THEN LET w=10
630 FOR i=1 TO 3: PRINT AT 21-y,5+x; FLASH 1; INK 7;CHR$(146+i): BEEP .05,.05
640 PRINT AT 21-y,5+x;" ";: NEXT i
670 IF w=3 THEN GO TO 740
690 LET f1=0: FOR i=1 TO n
700 IF x(i)=5+x AND y(i)=21-y THEN LET
x(i)=x(n): LET y(i)=y(n): LET h=h+1: LET
f1=1: LET s(i)=s(n): PRINT AT 21-y,4+x
;" "
710 NEXT i
720 IF f1 THEN LET n=n-1
740 IF n THEN GO TO 280
800 REM ***all destroyed***
801 BEEP .4,0: BEEP .5,4: BEEP .4,0: BEEP .5,5
810 LET t=FN x()-sb
820 PRINT AT 10,11; PAPER 0; INK 7; FLASH 1; BRIGHT 1;"GAME OVER"
830 PRINT INK 7;"you destroyed ";h;" ships."
835 PRINT "your time was ";t;" secs"
840 IF k THEN GO TO 880
850 BEEP .5,-2: BEEP .6,-6
860 STOP
880 INK 7: PRINT "PRESS THE SPACE BAR FOR A HARDER GAME.."
890 IF INKEY$<>" " THEN GO TO 890
900 IF h=9 THEN LET h=8: LET u=2*u: LET v=2*v
910 LET n=h+1: GO TO 40
920 PRINT AT y(k),x(k);" "
940 PRINT AT 3,0; INK 6; FLASH 1;"SHIP ESCAPES!!!";AT dy,dx;"*"
950 GO SUB 960: PRINT AT dy,dx;" ": GO SUB 960: NEXT i: LET k=0: GO TO 810
960 FOR j=1 TO 20: NEXT j: RETURN
980 DATA 147,144,140,129,147,136,161,160,130,143,143,141,161,160
990 DATA 2,3,7,3,9,3,6,4,4,10,5,5,7,10,

```

```

5,9,8,5
998 RETURN
999 STOP
1000 BORDER 0: PAPER 0: INK 6: CLS
1001 DEF FN x()=INT ((65536*PEEK 23674+256*PEEK 23673+PEEK 23672)/50)
1010 PRINT "A convoy of ships is sailing and you have to destroy them all"
1020 PRINT "using the gun (+) in the bottom corner of the screen."
1030 PRINT INVERSE 1;"Elevation"; INVERSE 0;" is measured in degrees(10-85) from the horizontal.": PRINT " To increase ";
1040 PRINT "press 8,": PRINT " To decrease press 2.": INK 7: PRINT " Maximum Range occurs when Elevation=45 degrees."
1050 PRINT : INK 6: PRINT INVERSE 1;"Bearing"; INVERSE 0;" is measured in degrees from -5 through 0(north) to 90(east).": PRINT " Increase by pressing 6."
1060 PRINT " Decrease by pressing 4."
1070 PRINT : PRINT " To fire press 5."
: PRINT : PRINT INVERSE 1;" Press enter to start the game ": INVERSE 0
1080 INPUT LINE a$: GO SUB 9000: BORDER 1: INK 1: PAPER 1: CLS : RETURN
1111 STOP
1999 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
9000 RESTORE 9010: FOR f=0 TO 7: READ a,b: POKE USR "a"+f,a: POKE USR "b"+f,b: NEXT f
9010 DATA 1,0,1,0,15,192,31,240,127,255,127,254,63,252,63,240
9020 FOR f=USR "c" TO USR "g": POKE f,INT (RND*256): NEXT f
9030 RETURN

```

# SEARCH AND RESCUE

TIM WARD

Original program by Geoffrey Little and Michael Orton

## GENERAL DESCRIPTION

A holiday cruiser has sunk off rocky coastline. It is your job to save the 20 survivors from lifeboats with your rescue helicopter. Skilfully fly down to a point above the boats where the survivors can be winched aboard, then fly behind the lighthouse and set them down safely on dry land. Keep an eye on your fuel level which decreases more rapidly if you are carrying passengers. However there is an unlimited fuel supply at the top of the lighthouse. Only after all 20 lifeboats have run aground will a score be given, along with an assessment of your ability.

## DETAILED DESCRIPTION

Lines 5-50 Define functions to determine position of helicopter.  
90-200 Winch up a survivor.  
201-215 Drop a passenger.  
216-222 Passenger is dashed to death on the rocks.  
224-230 Passenger is lost at sea.  
235-300 Passenger is winched to safety and runs into lighthouse.  
350-440 Helicopter stalls and explodes.  
450-925 Set up screen display of lighthouse and rocks.  
950-1001 Set up initial values of variables.  
1003-1030 Play the introductory tune and get user input for the helicopter controls.  
1031-1033 Pick up passenger and increase fuel consumption.  
1045-1053 Check for helicopter crash.  
1054-1065 Print lifeboats and choppy sea.

2000-3040 Print end page – give assessment and ask for another go.

7000-7160 Print instructions.

7500-8100 Set up user-defined graphics characters and create boat and wave strings.

```
1 CLS : CLEAR : RESTORE
5 REM ***Detect position of c
opter***
10 LET c=0: DEF FN a(a)=(CODE
(SCREEN$ (20,10))<>32)
20 DEF FN b(b)=(x>1 AND y<6 AN
D y>2 OR x>15 AND y<9 OR x>19)
25 DEF FN c(c)=(x>17 AND y>10)
26 LET d=0: DEF FN d(d)=(p=1 A
ND INKEY$=" ")
27 DEF FN e(e)=(fu-(4*f1))
50 GO SUB 7000: GO TO 890
90 REM ***Pick up survivor***
100 IF CODE (SCREEN$ (x+2,y))<>
32 THEN GO TO 106
105 RETURN
106 IF x>18 OR p=1 OR rp=0 THEN
RETURN
107 IF CODE b$(y-9)<>152 THEN
RETURN
110 PRINT AT x+1,y;CHR$ 160: PR
INT AT x,y; INK 2;h$(dir): PAUSE
8: PRINT AT x,y; INK 2;h$(dir+1
): PAUSE 8: PRINT AT x+1,y;" ";A
T x,y;" ": LET b$(y-9)=CHR$ 161
: PRINT AT x+2,y; INK 3;CHR$ 161
115 LET x=x-1: LET y=y-2: LET p
=1: RETURN
200 LET rr=0: IF x>=13 AND y<3
THEN LET rr=1
201 REM ***Drop passenger***
205 FOR j=1 TO 20: PRINT AT x+j
,y;CHR$ 160: PAUSE 5: PRINT AT x
+j,y;" "
210 IF x+j=16 AND y<3 AND rr=1
THEN GO TO 250
```



```

215 IF x+j=16 AND y<3 AND rr=0
THEN GO TO 300
216 REM ***Splat passenger***
220 IF x+j>=16 AND y>5 AND y<11
THEN PRINT AT x+j,y; INK 2;CHR
$ 156: BEEP .5,-15: PRINT AT x+j
,y;" ": LET dd=dd+1: LET p=0: LE
T rp=0: RETURN
222 IF x>=1 AND y>2 AND y<6 THE
N LET q$=SCREEN$ (x+1,y): PRINT
AT x+1,y; INK 2;CHR$ 156: BEEP
.5,-15: PAUSE 5: PRINT AT x+1,y;
q$: LET rp=0: LET p=0: LET dd=dd
+1: RETURN
224 REM ***Passenger falls into
sea***
225 IF x+j>19 AND y>10 THEN PR
INT AT (x+j),y;"SPLASH": PAUSE 5
: LET dd=dd+1: LET p=0: LET rp=0
: RETURN
230 NEXT j: LET p=0: LET rp=0:
RETURN
235 REM ***Passenger enter ligh
thouse***
250 PRINT AT x+j,y;" ": FOR s=1
TO 3
251 IF y+s>2 THEN LET p=0: LET
rp=0: LET su=su+1: RETURN
255 PRINT AT x+j,y+s;CHR$ 160:
PAUSE 5: PRINT AT x+j,y+s;" ": N
EXT s: LET su=su+1: LET p=0: LET
rp=0: RETURN
300 PRINT AT x+j,y; INK 2;CHR$
156: BEEP .5,-15: PAUSE 5: PRINT
AT x+j,y;" ": LET dd=dd+1: LET
p=0: LET rp=0: RETURN
350 REM ***Helicopter stalls &
explodes***
400 FOR w=1 TO 21
410 IF FN b(b) THEN PRINT PAP
ER 6;AT x,y;CHR$ 156;AT x,y+1;CH
R$ 156;AT x+1,y;CHR$ 156;AT x-1,

```

```

y;CHR$ 156: BEEP 1,-20: GO TO 20
00
420 LET x1=x: LET y1=y: LET x=x
+1
425 IF x1<>x OR y1<>y THEN PRI
NT AT x1,y1;" "
430 PRINT AT x,y; INK 2;h$(dir)
: BEEP .02,-40: BEEP .02,-40: PR
INT AT x,y; INK 2;h$(dir+1): PAU
SE 3: BEEP .02,-40: LET fu=FN e(
fu): IF fu<100 THEN FOR j=1 TO
3: BEEP .05,50: NEXT j
440 NEXT w
450 REM ***Set up screen***
890 LET r$=" "
895 PRINT AT 3,3;CHR$ 157;" ";C
HR$ 158;AT 2,4;CHR$ 159
900 FOR j=4 TO 16: PRINT AT j,3
;" "
910 IF j/4=INT (j/4) THEN PRIN
T AT j+1,3; INVERSE 1; BRIGHT 1;
" # ": LET j=j+1
920 NEXT j
925 FOR j=1 TO 5: PRINT AT 16+j
,0; INK 4;r$( TO 5+j);CHR$ 155:
NEXT j
950 REM ***Set variables***
1000 LET tot=20: LET fl=1: LET f
u=1000: LET rp=0: LET x=1: LET x
1=1: LET k=0: LET y=2: LET y1=2:
LET dir=1: LET p=0
1001 PRINT AT 20,10; INK 3;b$( T
O 21): PRINT AT 21,11; INK 1;w$
1003 REM ***Intro tune***
1004 FOR j=1 TO 2: BEEP .1,11: B
EEP .1,11: BEEP .8,16: BEEP .05,
11: BEEP .05,16: BEEP .05,11: BE
EP .05,16: BEEP 1,20: NEXT j
1005 LET q$=" ": LET dd=0: LET s
u=0: LET rr=0: LET f$=" "
1006 FOR v=1 TO 5
1010 IF INKEY$="Q" OR INKEY$="q"

```

```

THEN LET x=x-1
1011 IF x=1 AND y=3 AND INKEY$="
" THEN LET fu=fu+25
1012 IF fu>1000 THEN LET fu=100
0: LET y=y+1
1015 IF INKEY$="A" OR INKEY$="a"
THEN LET x=x+1
1020 IF INKEY$="O" OR INKEY$="o"
THEN LET y=y-1: LET dir=3
1021 IF INKEY$="P" OR INKEY$="p"
THEN LET y=y+1: LET dir=1
1022 LET k=k+1
1023 IF FN d(d) THEN GO SUB 200
1024 IF x<1 THEN LET x=1: LET d
ir=1
1025 IF p=1 THEN LET f1=1.5
1026 IF x>21 THEN LET x=21
1027 IF y<0 THEN LET y=0: LET d
ir=1
1028 IF y>30 THEN LET y=30: LET
dir=3
1029 IF x1<>x OR y1<>y THEN PRI
NT AT x1,y1;" "
1030 IF x<17 AND y>6 THEN GO TO
1049
1031 IF INKEY$=" " AND x>17 THEN
LET rp=1: REM ***Pick up passe
nger***
1032 IF p=1 THEN LET f1=1.5: RE
M ***Increase fuel if passenger
taken***
1033 IF p=0 THEN LET f1=1
1045 REM ***Detect crash***
1046 IF FN b(b) THEN PRINT PAP
ER 6:AT x,y:CHR$ 156:AT x,y+1:CH
R$ 156:AT x+1,y:CHR$ 156:AT x-1,
y:CHR$ 156: BEEP 1,-20: GO TO 20
00
1047 IF FN c(c) THEN GO SUB 100
1048 IF FN d(d) THEN GO SUB 200
1049 PRINT AT x,y: INK 2:h$(dir)
: BEEP .02,-40: BEEP .02,-40: PR

```

```

INT AT x,y: INK 2:h$(dir+1): PAU
SE 3: BEEP .02,-40: LET fu=FN e(
fu): IF fu<100 THEN FOR j=1 TO
3: BEEP .05,50: NEXT j
1050 LET rp=0: LET f$=STR$ (INT
(fu)): PRINT AT 0,28: INVERSE 1;
" ";AT 0,28:f$( TO ): IF fu<-1
0 THEN GO TO 400
1051 LET rp=0: LET x1=x: LET y1=
y: NEXT v
1052 IF INT (k/15)=k/15 THEN GO
TO 1055
1053 GO TO 1065
1054 REM ***Print boats & waves*
**
1055 LET b$=b$(2 TO )+" ": PRIN
T AT 20,10: INK 3;b$( TO 21)
1056 IF FN a(a) THEN PRINT AT 2
0,10:CHR$ 156: PAUSE 20: PRINT A
T 20,10;" "
1057 IF k>=1500 THEN GO TO 2100
1060 PRINT AT 21,11: INK 1:w$: L
ET w$=w$(2 TO )+w$(1)
1065 GO TO 1010
2000 CLS : PRINT AT 11,10:"You b
lew it!!": GO TO 3000
2100 CLS : PRINT AT 5,5:"End of
rescue operation."
2105 LET lost=tot-(dd+su)
2110 PRINT AT 7,5:"You saved ";s
u;(" people" AND su>1);(" person
" AND su=1);(CHR$ 8+" no-one" AN
D su=0): PRINT AT 9,5:lost;(" pe
ople were" AND lost>1);(" person
was" AND lost=1);(CHR$ 8+" no-o
ne was" AND lost=0);" lost at se
a."
2120 IF dd=0 THEN GO TO 2140
2130 PRINT AT 11,5:"Unfortunately
";dd;(" people" AND dd>1);(" pe
rson" AND dd=1);" died": PRINT A
T 12,5;"due to your incompetence

```

```

.: PRINT AT 13,5;"You are suspe
nded pending      an enquiry."
2140 GO TO 3000
3000 PRINT : PRINT "      Ano
ther try Y/N"
3010 LET a$=INKEY$: IF a$="" THE
N GO TO 3010
3020 IF a$="N" OR a$="n" THEN S
TOP
3030 IF a$="Y" OR a$="y" THEN R
UN
3040 GO TO 3010
7000 CLS : PRINT AT 1,7; INVERSE
1; FLASH 1;" Air Sea Rescue ":
PRINT AT 3,0;"You are the pilot
of a rescue helicopter,based o
n a lighthouse located on a dange
rous part of the coast.";
7010 PRINT "A holiday cruiser ha
s sank off your coast,the crew a
nd passengers have got away in th
e lifeboats. ";
7020 PRINT "However the lifeboat
s are being carried by the curren
t onto the rocks!!"
7030 PRINT : PRINT "It is your j
ob to save as many passengers a
nd crew as possible.You must fly
down to each boat and pick up
a survivor."
7040 PRINT "Don't get too close
to the boat the downdraft from y
our rotors will prevent the su
rvivor from standing up."
7050 GO SUB 8000
7051 PRINT #1;AT 1,0;"Press any
key to continue": IF INKEY$="" T
HEN GO TO 7051
7060 CLS : PRINT AT 1,6;"Flying
Instructions"
7070 PRINT : PRINT "To fly the h
elicopter use the following ke

```

```

ys:-"
7080 PRINT AT 6,10;"Q.....up";AT
8,10;"A.....down";AT 10,10;"O..
...left";AT 12,10;"P.....right"
7090 PRINT : PRINT "The SPACE ke
y has two functions.Press it to
collect a survivor."
7100 PRINT "[The helicopter must
be above the mast of the boat
but not too close]."
7110 PRINT #1;AT 1,0;"Press any
key to continue": IF INKEY$="" T
HEN GO TO 7110
7120 CLS : PRINT "Pressing the S
PACE key releases the pasenger."
: PRINT "You must land the passe
nger behind the lighthouse b
ut do not release him higher than
the low-est window on the light
house."
7130 PRINT : PRINT "Your fuel st
atus is indicated on the right
,you may refuel by positioning
the helicopter directly abo
ve the top of the lighthouse a
nd pressing the SPACE key (n
ot with a passenger aboard thoug
h)."
7140 PRINT : PRINT "When your ta
nk is full or when you release
the SPACE key - refueling wi
ll stop."
7150 PRINT #1;AT 1,0;"Press the
SPACE key to begin": IF INKEY$=""
THEN GO TO 7150
7160 CLS : GO TO 890
7500 REM ***Set up user defined
graphics and create boat & wave
strings***
8000 FOR j=1 TO 18: READ a
8010 FOR k=0 TO 7: READ b
8020 POKE USR CHR$ a+k,b

```



```

8025 NEXT k: NEXT j
8030 DATA 144,254,128,240,152,25
2,248,144,254,145,0,128,240,152,
252,248,144,254
8035 DATA 146,0,0,3,135,157,127,
0,3,147,127,0,3,135,157,127,0,3
8040 DATA 148,0,0,192,225,190,22
4,0,192,149,252,0,192,225,190,22
4,0,192
8045 DATA 150,127,1,15,31,49,31,
9,127,151,1,1,15,31,49,31,9,127
8046 DATA 152,1,1,1,129,127,63,3
1,15,153,0,0,0,0,255,254,248,240
8047 DATA 154,96,48,56,124,255,2
55,255,255,155,4,24,120,111,140,
231,255,255
8048 DATA 156,157,130,68,1,242,4
,136,145,157,0,0,255,139,139,139
,139,139,158,0,0,255,17,17,17,17
,17,159,24,24,24,60,126,255,255,
255
8049 DATA 160,56,16,124,186,145,
168,68,198,161,0,0,0,129,127,63,
31,15
8090 DIM h$(5,2): LET h$(1,1 TO
)=CHR$ 146: LET h$(1,2 TO )=CHR$
144: LET h$(2,1 TO )=CHR$ 147:
LET h$(2,2 TO )=CHR$ 145
8095 LET h$(3,1 TO )=CHR$ 150: L
ET h$(3,2 TO )=CHR$ 148: LET h$(
4,1 TO )=CHR$ 151: LET h$(4,2 TO
)=CHR$ 149
8098 DIM b$(100)
8100 FOR j=1 TO 20: LET w=INT ((
RND*89)+10): LET b$(w)=CHR$ 152:
LET b$(w+1)=CHR$ 153: NEXT j
8110 DIM w$(21): FOR j=1 TO 21 S
TEP 2: LET w$(j TO )=CHR$ 154+CH
R$ 155: NEXT j: LET w$(9)=CHR$ 1
54: LET w$(10)=CHR$ 155
8115 RETURN

```

## HINTS AND TIPS

### KEEPING TRACK OF MEMORY

It is often useful for a program to be able to tell how much memory is available to it; an adventure game might set up more rooms in a 48K Spectrum than in a 16K model, or a music synthesiser program could find out how many notes it could store.

PEEK 23733 returns 127 on a 16K Spectrum and 255 if you have 48K.

The number of free bytes remaining on the system is given by 65536-USR 7962 while if you wish to know the length of your program you can use PRINT PEEK 23627 - PEEK 23635 + 256 \* (PEEK 23628 - PEEK 23636)

### SOME USEFUL POKES

Here is a collection of POKES which you may find useful.

**POKE 23561,n** This gives the time for which a key must be held down, in fiftieths of a second, before auto-repeat starts. Initially this is set to 35.

**POKE 23652,n** This is the time between repeats of a key, initially 5.

**POKE 23609,n** This varies the keyboard click length. Increasing its value to about 20 produces a more audible beep.

**POKE 23692,255**. Scroll counter. POKEing this with 255 during a program stops the computer asking 'scroll?' when the screen is full.

### DISABLING BREAK/TRAPPING ERRORS

One of the most sought-after routines for the Spectrum is a method of disabling the BREAK key. Even though Sinclair announced this could be done when the computer was launched, the manuals don't mention any technique.

You can do this with POKE 23613, PEEK 23730-5 to disable BREAK, and POKE 23613, PEEK 23730-3 to re-enable it. If you are sure that RAMTOP is in its usual place, you can change the pokes to 82 and 84 respectively.

The POKE works by altering the low byte of system variable

ERR SP, the address to which the stack is reset after an error. Normally this points to a routine that terminates program execution, but the POKE makes it point to a routine that continues execution.

Using this POKE traps all errors except 'Nonsense in Basic'. You can find out which error occurred with PEEK 23610. If you do this then clear any previous errors first with POKE 23610,255.

There are several drawbacks to using this method so careful note should be taken of the following points:-

The POKE only works 99% of the time: location 23614 should strictly be altered too.

When an error occurs, any further statements on a multi-statement line are skipped.

Calling a subroutine will clear the error trap.

The machine will crash if program execution ends before error trapping is turned off, a 'Nonsense in Basic' error occurs, or you attempt to use the POKE inside a subroutine.

## SETTING AND RESETTING CAPS LOCK

It is sometimes useful to be able to switch the Spectrum keyboard's caps mode on and off, so as to control the keys which a program is testing for. Caps mode is controlled by the system variable FLAGS2 at location 23658. POKE 23658,8 sets caps mode, but also resets all the other flags in that register.

There is a ROM routine which can be called using RANDOMIZE USR 4317 but if caps mode is already set, this call will reset it.

The solution is a short machine code routine which can be loaded anywhere in memory. It consists of just three instructions:-

```
LD HL,23658
```

```
SET 3,(HL)
```

```
RET
```

A suitable program to set up this routine would like like this:-

```
1000 LET start= any suitable address
```

```
1010 FOR a=start TO start+5
```

```
1020 READ b: POKE a,b
```

```
1030 NEXT a
```

```
1040 DATA 33,106,92,203,222,201
```

Caps mode can then be set by RANDOMIZE USR (start)

To force the Spectrum into lower case mode change 222 to 158 in the data in line 1040.

## REDIRECTING PRINTOUT

Printing on the Spectrum is directed through a series of streams. These are accessed using the # symbol, so PRINT#3;"Hello" would print 'Hello' on the printer. Printout can be sent to the bottom two lines of the screen by using:-

```
PRINT#0;"message";
```

Try INPUT#2;"Enter a number ";#0;a This will print the prompt in the upper part of the screen. The input stream must be set to #0 after printing the prompt or an error message results.

The command OPEN# can be used to change output streams permanently. OPEN#2,"k" will attempt to send all screen output to the keyboard (device 'k'), but this results in it appearing in the lower part of the screen. Care must be taken not to print more than twenty lines without clearing the screen or an 'Out of screen' error will result.

To send all screen output to the printer use OPEN#2,"p" and to send printer output to the screen type OPEN#3,"s".

## PRODUCING FLASHY LISTINGS

It is not widely known that Spectrum listings can incorporate colour commands (and indeed OVER and BRIGHT) as well as inverse video. This feature is probably a happy accident of the machine's logical structure, but it allows you to produce pretty listings.

The way to obtain these effects is to go into extended mode (E - cursor) after the line number. You can do this on first entering the line, or when using EDIT. You then use the number keys, shifted and unshifted, to get the effects. These keys are shown on page 115 of the User Guide.

This capability has several uses:-

- (i) Producing an invisible listing (even when EDIT is used)
- (ii) Drawing attention to an important REM statement or a test line which must be deleted after debugging
- (iii) Highlighting all the lines within a FOR - NEXT loop.

The effect of a control code is not confined to a single line, so if you wish to highlight a particular line, you must include two changes - one to produce the effect, and another to restore the original conditions. Each of these control codes takes up two bytes of memory and very strange effects can be caused by deleting just one of the bytes.





```

1020 FOR p=15744 TO 15823
1030 POKE c,PEEK p: POKE c+1,PEEK p
1040 LET c=c+2: NEXT p
1050 RETURN
2000 REM print x in double height
2010 LET a$=STR$ INT ABS x
2020 FOR a=0 TO 1
2030 FOR p=1 TO LEN a$
2040 LET z=CODE (a$(p))-48
2050 PRINT CHR$ (144+a+2*z);
2060 NEXT p: PRINT
2070 NEXT a
2080 RETURN

```

## GETTING THE MOST OUT OF DRAW

The Sinclair manual documents two features of the DRAW command. The first uses two parameters to draw a straight line relative to the current screen position. The second adds another parameter, an angle in radians, and draws a curve, turning through the given angle.

To obtain an absolute draw (i.e. drawing a line to a specified screen position from the current plot position) is quite simple. Memory locations 23677 and 23678 contain respectively the x and y co-ordinates of the last point plotted so the statement-  
DRAW x-PEEK 23677, y-PEEK 23678

would draw a straight line to position x,y.

By making the third parameter in a draw statement very large, some strange and useful effects can be achieved. The following statement produces a shaded circle:-

PLOT 127,87: DRAW 10,10,778

Some more values to try are 555,991,8440,6974, and 1871, but you should be able to find many more. Try using large values with decimals or setting OVER 1.

## GRAPHICS FUNCTIONS FOR PASCAL

Hisoft's Pascal IV compiler produces code which runs many times faster than an equivalent Basic program, but does not offer any graphics facilities. This program provides several graphics features as follows:-

CIRCLE(X,Y,R) draws a circle, centre X,Y and radius R

DISC(X,Y,R) draws a series of circles at centre X,Y in an attempt

to fill a disc of radius R

PLOT(X,Y) sets the point X,Y

DRAWBY(X,Y) is a relative draw like the Spectrum's DRAW function, but no third parameter is allowed, so curves cannot be drawn.

LINE(X1,Y1,X2,Y2) draws a straight line between X1,Y1 and X2,Y2.

FILL(X1,Y1,X2,Y2) shades a rectangle. X1,Y1 specifies the bottom left hand corner and X2,Y2 is the top right corner.

COPY is identical to the Basic COPY function.

Most of the functions work by calling the appropriate ROM subroutines. Others use a function which has already been defined.

```

Run??
AF92 210 PROCEDURE CIRCLE(X,Y,R
: INTEGER);
AF95 220 BEGIN
AFAD 230 POKE (23681,ORD(R));
AFBC 240 POKE (23728,ORD(X));
AFCD 250 POKE (23729,ORD(Y));
AFDA 260 INLINE (#09,#E5,#09,#3A
, #B0, #5C, #CD, #28, #2D, #3A, #B1, #5C
, #CD, #28, #2D, #3A, #B1, #5C);
AFEC 270 INLINE (#CD, #28, #2D, #CD
, #2D, #23, #09, #E1, #09);
AFF5 280 END;
AFFF 290
AFFF 300
AFFF 310 PROCEDURE DISC(X,Y,R: I
NTEGER);
B002 320 VAR I: INTEGER;
B002 330 BEGIN
B01A 340 FOR I:=R DOWNT0 1 DO
B03F 350 CIRCLE(X,Y,I);
B063 360 END;
B06E 370
B06E 380
B06E 390 PROCEDURE PLOT(A,B: INT
EGER);
B071 400 BEGIN
B089 410 POKE (23728,ORD(A));
B098 420 POKE (23729,ORD(B));
B0A7 430 INLINE (#ED, #4B, #B0, #5C
, #CD, #E5, #22);
B0AE 440 END;
B0B8 450
B0B8 460 PROCEDURE DRAWBY(X,Y: I
NTEGER);
B0BB 470 BEGIN
B0D3 480 IF X<0 THEN POKE (23296
,ORD(#FF));
B0F6 490 IF X>=0 THEN POKE (2329
6,ORD(1));
B11B 500 IF Y<0 THEN POKE (23297
,ORD(#FF));
B13E 510 IF Y>=0 THEN POKE (2329
7,ORD(1));
B163 520 POKE (23728,ORD(ABS(X)
));
B175 530 POKE (23729,ORD(ABS(Y)
));

```

```

B187 540 INLINE (#D9,#E5,#D9,#ED
, #4B,#B0,#5C,#ED,#5B,#00,#5B,#CD
, #BA,#24,#D9,#E1,#D9);
B198 550 END;
B1A2 560
B1A2 570
B1A2 580 PROCEDURE LINE (X1,Y1,X
2,Y2: INTEGER);
B1A5 590 BEGIN
B1B0 600 POKE (23677,ORD (X1));
B1CC 610 POKE (23678,ORD (Y1));
B1DB 620 DRAWBY (X2-X1,Y2-Y1);
B210 630 END;
B21A 640
B21A 650
B21A 660 PROCEDURE FILL (X1,Y1,X
2,Y2, INK: INTEGER);
B21D 670 VAR I: INTEGER;
B21D 680 BEGIN
B235 690 WRITELN (CHR (16),CHR (IN
K));
B249 700 FOR I:=X1 TO X2 DO
B276 710 BEGIN
B279 720 PLOT (I,Y1);
B290 730 DRAWBY (0,Y2-Y1);
B2B3 740 END;
B2B7 750 END;
B2C2 760
B2C2 770
B2C2 780 PROCEDURE COPY;
B2C5 790 BEGIN
B2DD 800 USER (#0EAC);
B2E3 810 END;

```

## GENERATING LOOK-UP TABLES

Spectrum Basic includes the READ, DATA and RESTORE instructions. The manual only shows RESTORE followed by a line number, but it will in fact accept an expression. This provides the basis for a fast and economical look-up table facility. The following example illustrates the method:-

```

50 REM month table look-up
101 DATA 31,"January"
102 DATA 28,"February"
103 DATA 31,"March"
104 DATA 30,"April"
105 DATA 31,"May"
106 DATA 30,"June"
107 DATA 31,"July"
108 DATA 31,"August"
109 DATA 30,"September"
110 DATA 31,"October"
111 DATA 30,"November"
112 DATA 31,"December"
200 INPUT "Month number";mm

```

```

300 RESTORE mm+100
400 READ dd,a$
500 PRINT "There are ";dd;" day
s in ";a$
600 GO TO 200

```

The line numbers of the DATA statements are arranged so that a simple expression can relate the required item of DATA to the month number. Line 300 uses this expression to set the read pointer so that line 400 retrieves the corresponding days in the month and month name.

This technique is especially useful with tables containing mixed data types and variable length strings. It offers a considerable memory saving over the usual method of setting up a series of arrays.

## LOGICAL FUNCTIONS

The AND, OR and NOT operators in Spectrum Basic are different from their equivalents in most other micros in that they do not operate in a bit by bit manner - a facility which would be useful in many applications, such as handling I/O ports. The routine shown below provides logical functions for bit handling. The machine code is placed into the space reserved for user-defined graphics 'n' to 'u', but the routine can easily be adapted to place the code elsewhere.

The 'and', 'or' and 'xor' functions each require 14 bytes of code, while 'not' requires 12 bytes. The machine code accesses the parameters required by the functions via the address given in system variable DEFADD at location 23563, this being where the parameters of any currently executing FN are placed.

The functions provided are as follows:-

FN(a,b) returns a and b

FNo(a,b) returns a or b

FNx(a,b) returns a xor b

FNn(a) returns not a

The functions will only yield sensible values if the parameters supplied are in the range 0 - 255.

```

9000 REM install logic functions
9010 RESTORE 9240
9020 REM code common to and,or and xor

```



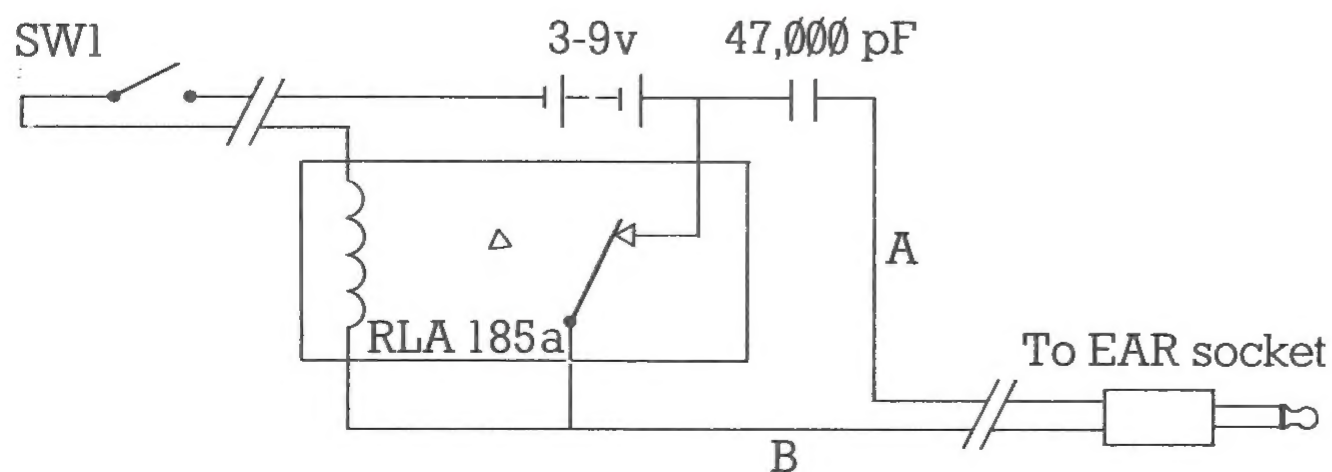
```

9030 FOR a=0 TO 14
9040 READ byte
9050 POKE USR "p"+a,byte
9060 POKE USR "r"+a,byte
9070 POKE USR "t"+a,byte
9080 NEXT a
9090 REM poke in opcodes
9100 POKE USR "p"+10,160
9110 POKE USR "r"+10,176
9120 POKE USR "t"+10,168
9130 REM code for not
9140 FOR a=0 TO 11
9150 READ byte
9160 POKE USR "n"+a,byte
9170 NEXT a
9180 REM FN definitions
9190 DEF FN a(a,b)=USR USR "p"
9200 DEF FN o(a,b)=USR USR "r"
9210 DEF FN x(a,b)=USR USR "t"
9220 DEF FN n(a)=USR USR "n"
9230 DATA 221,42,11,92,221,126,4
,221,70,12,0,79,6,0,201
9240 DATA 221,42,11,92,221,126,4
,47,79,6,0,201
9250 RETURN

```

### SWITCH TESTING PROGRAM

This is a simple circuit which allows the Spectrum to detect whether a switch is on or off.



The wires A & B are connected to the normally closed contacts of the relay RLA. Closing SW1 causes the relay to switch on and

off rapidly, sending a series of pulses to the computer's EAR socket. The capacitor suppresses the relay and limits the circuit current. SW1 can be any normally open switch; a thermostat or a burglar alarm pressure mat for example. The following program is necessary to use the switch:-

```

10 GO SUB 5000: CLEAR
20 LET a=USR USR "r"
30 LET k=PEEK 23608
40 IF k<3 THEN GO TO 20
50 REM the switch is now on
60 REM rest of program
5000 FOR n=USR "r" TO USR "u"
5010 READ a
5020 POKE n,a
5030 NEXT n
5040 DATA 33,56,92,62,0,119,6,25
5,219,250,254,255,40,1,52,16,247
,201,0,0,0,0,0,0,0
5050 RETURN

```

The machine code routine occupies the memory for user defined graphics 'r' to 'u' but if these are needed, it can be moved to another address by changing line 5000.

### INDELIBLE COPYRIGHT MESSAGE

You can create a program line which cannot be deleted or edited by normal methods, using a POKE statement. This is useful for adding copyright messages to your programs. Type in the line you wish to protect as line 1 then protect it by typing:-  
 POKE (PEEK 23635 + 256 \* PEEK 23636) + 1,0  
 This creates a line 0 which will run normally but is not otherwise recognised by the system.

### GENERATING ERROR MESSAGES

One of the reasons a machine code user may want to display a message on the screen of the Spectrum is that an error has been made somewhere. In many cases, one of the Sinclair standard error codes could be used. For instance, in some assembler programs an attempt to load a single register with a number greater than 255 could be met by report code B - 'Integer out of



range', which can be followed by a line number and statement number of the programmer's choice.

The trick is to use a ROM subroutine at 08h. The machine code command RST 8 causes a return to Basic, generating an error message. The byte following the RST command should contain the number one less than the error code wanted, and addresses 5C45h and 5C46h (23621/2 dec.) should contain the low and high bytes respectively, of the line number. Address 5C47h (23623 dec.) contains the statement number. For example:-

```
LD HL,1234      21 D2 04
LD (23621),HL   22 45 5C
LD A,5          3E 05
LD (23623),A    32 47 5C
RST 8           CF
19             13
```

The short program above will give the message 'K Invalid colour 1234:5'

If you look in the Sinclair manual, only codes 0-9 and A-R are listed. Other values give some interesting reports. Error T, for instance, is the Sinclair copyright message. Type in the following two line program and run it. You will get error code 1. Type NEXT n and press ENTER, and you will get error 2. In this way you can work through all 256 codes.

```
10 FOR n=0 TO 255
20 POKE 23610,n
```

## WHITE NOISE GENERATOR

Many computers have a facility for generating white noise. This is useful for simulating explosions and rocket burners etc.

The program below produces white noise by means of a machine code routine. The duration of the sound can be set with POKE 32583,n and the pitch can be changed by POKEing 32589. The routine can be called with RANDOMIZE USR 32581.

48 K users can load the code higher up in memory without making any changes to it, but the POKE addresses for duration and pitch will of course change accordingly.

```
10 CLEAR 32580
20 FOR p=32581 TO 32597
30 READ a: POKE p,a
40 NEXT p
```

```
50 DATA 33,0,10,43,126,211,254,6,7,5,32,-3,175,132,200,24,-14
100 REM Demonstration
110 BEEP .007,19: PAUSE 10: BEEP .007,31: PAUSE 10:
RANDOMIZE USR 32581: PAUSE 5: RANDOMIZE USR 32581:
PAUSE 5: BEEP .007,19: PAUSE 10: BEEP .007,31: PAUSE 10:
RANDOMIZE USR 32581: PAUSE 5: BEEP .007,31: PAUSE 10
120 GOTO 110
```

## ADDING NOISE TO PROGRAMS

The Spectrum's BEEP command is controlled by the machine's CPU (central processing unit), so giving out continuous white noise while other things are happening on the screen is difficult.

However, it may be useful to know that you can produce a series of short clicks by accessing the output port which controls border colour and the computer's speaker. Try this program:-

```
10 LET bd=PEEK 23624/8: REM 23624 holds current border col.
20 OUT 254, bd+16: OUT 254,bd
30 REM action routine
40 DRAW INK RND*8;RND*255-PEEK 23677,RND*175-PEEK 23678
50 GOTO 20
```

